
AlliGator

Release 0.86

X. Michalet

Apr 11, 2024

CONTENTS

1	Contents	1
1.1	How to use this manual	1
1.1.1	Notations	1
1.2	AlliGator Overview	2
1.3	AlliGator Windows	2
1.3.1	Main windows	2
1.3.2	Accessory windows	2
1.4	AlliGator Panels	3
1.4.1	Source Image Panel	3
1.4.2	Image Histogram Panel	8
1.4.3	Fluorescence Decay Panel	12
1.4.4	Fluorescence Decay Statistics Panel	15
1.4.5	Intensity Time Trace Panel	15
1.4.6	Phasor Plot Panel	15
1.4.7	Phasor Graph Panel	15
1.4.8	Lifetime & Other Parameters Analysis Panel	15
1.4.9	Phasor Ratio Analysis Panel	15
1.4.10	Decay Fit Parameter Map Panel	15
1.5	AlliGator Menus	15
1.5.1	File Menu	15
1.5.2	Windows Menu	16
1.5.3	Scripts Menu	17
1.5.4	Analysis Menu	17
1.5.5	Calibration Menu	19
1.5.6	Settings Menu	19
1.5.7	Help Menu	20
1.6	Alligator User Guide	20
1.6.1	Loading & Saving FLI Datasets	20
1.6.2	Loading & Saving FLI Dataset Series	23
1.6.3	Regions of Interest	25
1.6.4	Decay Pre-processing	34
1.6.5	Fluorescence Decay Fitting	39
1.7	Alligator Reference	42
1.7.1	AlliGator HDF5 File Format	42
1.7.2	Python Plugins	54
1.7.3	AlliGator Python Plugins API v 0.1	63
1.7.4	Graph Object	63
1.7.5	Image Color Palette	68
1.7.6	Selected Plot Style Window	71
1.8	AlliGator Tutorials	72

1.9	AlliGator Shortcuts	72
1.10	AlliGator Installation	73
1.10.1	System Requirements	73
1.10.2	Download the installer(s)	73
1.11	AlliGator Version History	75
1.11.1	0.86 (2024-04-11)	76
1.11.2	0.85.1 (2024-04-04)	76
1.11.3	0.85 (2024-03-31)	76
1.11.4	0.84 (2024-03-08)	77
1.11.5	0.83 (2024-02-27)	77
1.11.6	0.82.1 (2024-02-14)	77
1.11.7	0.82 (2024-02-07)	77
1.11.8	0.81 (2024-02-03)	78
1.11.9	0.80 (2024-01-16)	78
1.11.10	0.79.2 (2024-01-05)	79
1.11.11	0.79.1 (2024-01-04)	79
1.11.12	0.79 (2024-01-04)	79
1.11.13	0.78 (2023-12-22)	80
1.11.14	0.77 (2023-12-16)	80
1.11.15	0.76 (2023-12-03)	81
1.11.16	0.75.2 (2023-11-26)	81
1.11.17	0.75.1 (2023-11-22)	81
1.11.18	0.75 (2023-11-19)	82
1.11.19	0.74 (2023-11-12)	82
1.11.20	0.73 (2023-09-18)	83
1.11.21	0.72 (2023-08-10)	83
1.11.22	0.71 (2023-06-26)	84
1.11.23	0.70 (2023-05-30)	85
1.11.24	0.69 (2023-04-17)	85
1.11.25	0.68.1	86
1.11.26	0.68 (2023-03-05)	86
1.11.27	0.67 (2023-01-23)	87
1.11.28	0.66 (2022-12-31)	88
1.11.29	0.65 (2022-12-15)	88
1.11.30	0.64 (2022-12-04)	89
1.11.31	0.63 (2022-11-26)	89
1.11.32	0.62.2 (2022-11-08)	90
1.11.33	0.62.1 (2022-11-04)	90
1.11.34	0.62 (2022-11-02)	90
1.11.35	0.61 (2022-10-23)	90
1.11.36	0.60.4 (2022-08-18)	91
1.11.37	0.60.3 (2022-08-15)	91
1.11.38	0.60.2 (2022-08-13)	91
1.11.39	0.60.1 (2022-08-04)	92
1.11.40	0.60 (2022-08-03)	92
1.11.41	0.59 (2022-08-01)	92
1.11.42	0.58 (2022-07-28)	93
1.11.43	0.57 (2022-07-21)	93
1.11.44	0.56.4 (2022-07-17)	93
1.11.45	0.56.3 (2022-06-30)	93
1.11.46	0.56.2 (2022-06-29)	94
1.11.47	0.56.1 (2022-06-27)	94
1.11.48	0.56 (2022-06-26)	94
1.11.49	0.55 (2022-05-13)	95

1.11.50	0.54 (2022-05-12)	95
1.11.51	0.53.1 (2022-05-10)	96
1.11.52	0.53 (2022-05-07)	96
1.11.53	0.52 (2022-05-05)	96
1.11.54	0.51 (2022-03-07)	97
1.11.55	0.50 (2022-01-31)	97
1.11.56	0.49 (2022-01-30)	97
1.11.57	0.48.4 (2022-01-14)	98
1.11.58	0.48.3 (2022-01-14)	98
1.11.59	0.48.2 (2022-01-11)	98
1.11.60	0.48.1 (2022-01-08)	98
1.11.61	0.48 (2022-01-07)	98
1.11.62	0.47 (2021-12-18)	99
1.11.63	0.46 (2021-12-11)	99
1.11.64	0.45.2 (2021-12-03)	100
1.11.65	0.45.1 (2021-12-01)	100
1.11.66	0.45 (2021-11-30)	100
1.11.67	0.44.3 (2021-11-12)	100
1.11.68	0.44.2 (2021-11-11)	101
1.11.69	0.44.1 (2021-11-07)	101
1.11.70	0.44 (2021-11-06)	101
1.11.71	0.43 (2021-11-03)	102
1.11.72	0.42.3 (2021-10-29)	102
1.11.73	0.42.2 (2021-10-27)	102
1.11.74	0.42.1 (2021-10-21)	102
1.11.75	0.42 (2021-10-158)	102
1.11.76	0.41 (2021-10-10)	103
1.11.77	0.40.1 (2021-10-08)	103
1.11.78	0.40 (2021-10-07)	103
1.11.79	0.39.4 (2021-10-06)	103
1.11.80	0.39.3 (2021-09-30)	104
1.11.81	0.39.2 (2021-09-29)	104
1.11.82	0.39.1 (2021-09-28)	104
1.11.83	0.39 (2021-09-27)	104
1.11.84	0.38.1 (2021-09-14)	104
1.11.85	0.38 (2021-09-13)	105
1.11.86	0.37 (2021-08-16)	106
1.11.87	0.36 (2021-08-05)	106
1.11.88	0.35 (2021-07-26)	106
1.11.89	0.34.4 (2021-07-23)	107
1.11.90	0.34.3 (2021-07-22)	107
1.11.91	0.34.2 (2021-07-22)	107
1.11.92	0.34.1 (2021-07-01)	107
1.11.93	0.34 (2021-06-28)	107
1.11.94	0.33.7 (2021-06-27)	108
1.11.95	0.33.6 (2021-06-27)	108
1.11.96	0.33.5 (2021-06-26)	108
1.11.97	0.33.3 (2021-06-24)	108
1.11.98	0.33.2 (2021-06-24)	108
1.11.99	0.33 (2021-06-22)	109
1.11.1000.32	(2021-04-01)	111
1.11.1010.31	(2021-01-29)	111
1.11.1020.30	(2020-08-12)	111
1.11.1030.29	(2020-08-11)	112

1.11.1040.28 (2020-08-03)	112
1.11.1050.27 (2020-07-06)	112
1.11.1060.26 (2020-06-28)	113
1.11.1070.25 (2020-06-22)	113
1.11.1080.24 (2020-06-12)	113
1.11.1090.23.1 (2020-05-19)	114
1.11.1100.22 (2020-05-17)	114
1.11.1110.21.1 (2020-05-11)	114
1.11.1120.21 (2020-05-10)	114
1.11.1130.20.4 (2019-11-26)	115
1.11.1140.20.3 (2019-11-21)	115
1.11.1150.20.2 (2019-11-04)	116
1.11.1160.20.1 (2019-10-23)	116
1.11.1170.20 (2019-10-05)	116
1.11.1180.19 (2019-09-15)	117
1.11.1190.18.2 (2019-09-13)	117
1.11.1200.18.1 (2019-08-05)	117
1.11.1210.18 (2019-06-19)	117
1.11.1220.18 beta* (2019-06-03)	118
1.11.1230.17.5 (2019-05-03)	118
1.11.1240.17.4 (2019-04-04)	119
1.11.1250.17.3 (2019-03-22)	119
1.11.1260.17.2 (2019-03-12)	119
1.11.1270.17.1 (2019-03-12)	119
1.11.1280.17 (2019-03-10)	120
1.11.1290.16.2 (2019-03-01)	120
1.11.1300.16.1 (2019-02-28)	120
1.11.1310.15.2 (2019-02-19)	121
1.11.1320.15.1 (2019-02-17)	121
1.11.1330.15 (2019-02-09)	121
1.11.1340.13 (2018-09-18)	122
1.11.1350.12 (2018-03-29)	122
1.11.1360.10.11 (2018-03-13)	122
1.11.1370.10.10 (2018-03-11)	122
1.11.1380.10.9 (2018-02-21)	122
1.11.1390.10.8 (2018-02-10)	123
1.11.1400.9.13 (not released)	123
1.11.1410.9.12 (2017-09-08)	124
1.11.1420.9.11 (2017-08-22)	125
1.11.1430.9.10 (2017-05-17)	125
1.11.1440.9.9 (2017-03-26)	125
1.11.1450.9.8 (2017-01-22)	125
1.11.1460.9.7 (2017-01-13)	125
1.11.1470.9.6 (2017-01-03)	126
1.11.1480.9.5 (2016-12-14)	126
1.11.1490.9.4 (2016-12-01, updated 2016-12-08)	126
1.11.1500.9.3 (2016-10-29)	126
1.12 Bibliography	127
1.13 Acknowledgments	127
1.14 AlliGator Support	128
1.14.1 How to report an issue	128

CONTENTS

1.1 How to use this manual

The manual is work in progress and is intended to replace the legacy manual hosted on Google Sites at <https://web.archive.org/web/20201028034436/https://sites.google.com/a/g.ucla.edu/alligator/manual>. Missing information in this manual can in general be found in that legacy manual.

The manual is (generally) updated as new versions of the software are released. The version you are looking at is indicated in the small v: `X.XX` drop-down menu on the bottom right of the Table of Content column in the online manual.

Change this to the version you want to learn about, if that is not the latest version.

It is also possible to download a PDF version of the manual by selecting PDF in the aforementioned drop-down menu.

You can click on links embedded in the manual's pages to jump directly to a topic of interest, or select a topic in the Table of Content.

Access the Table of Content by clicking on the Table of Contents link on the left or on the AlliGator x.xx.x documentation link at the top.

Alternatively, you can simply follow the page hierarchy by clicking on the Previous or Next buttons at the bottom of each page.

It is of course also possible to use the Go back or Go forward buttons of the browser to step through the browsing history.

1.1.1 Notations

Throughout this manual, individual objects (*controls*, which the user can interact with, or *indicators*, which only display information to the user) will be referred to by their name (which shows up in the **Context Help** window when opened) highlighted in *italics*.

Windows or panels will be indicated by their title indicated in **bold**. For windows with subpanels, the notation **Window:Subpanel** will be used.

Values which a control or indicator can take will be indicated with `this` style.

Menus and menu items will be rendered as follows: `menu item` or `menu:submenu:menu item`.

Links to different sections or pages of the manual will appear underlined when moving the mouse over them, like the link to [this page](#)

1.2 AlliGator Overview

AlliGator is a software for fluorescence lifetime image data analysis using the phasor approach and standard nonlinear fitting.

It was initially developed as part of a collaborative project described in ref. [Chen2018] listed on the [Bibliography page](#).

If you are using AlliGator in a publication or otherwise, please consider citing this reference.

1.3 AlliGator Windows

There are 3 main windows in AlliGator's GUI and several accessory windows listed below. Most can be accessed directly from the Windows menu (see [Windows Menu](#)).

1.3.1 Main windows

- alligator-main-window
- alligator-notebook-window
- alligator-settings-window

1.3.2 Accessory windows

- alligator-about-window
- alligator-help-window
- alligator-dataset-information-window
- alligator-source-image-roi-manager-window
- alligator-phasor-plot-roi-manager-window
- alligator-phasor-harmonic-manager-window
- alligator-image-profile-window
- alligator-local-decay-graph-window
- alligator-histogram-window
- alligator-plot-editor-window

AlliGator's main window's functionalities are described in the AlliGator manual pages. The Source Image ROI Manager Window is discussed in [alligator-source-image-roi-manager-window](#) and [Source Image Panel](#). The Phasor Plot ROI Manager Window is discussed in [alligator-phasor-plot-roi-manager-window](#) and [Phasor Plot Panel](#). The Phasor Harmonic Manager Window is discussed in [phasor-harmonic-analysis](#). The other windows' functions are described in their respective manual pages.

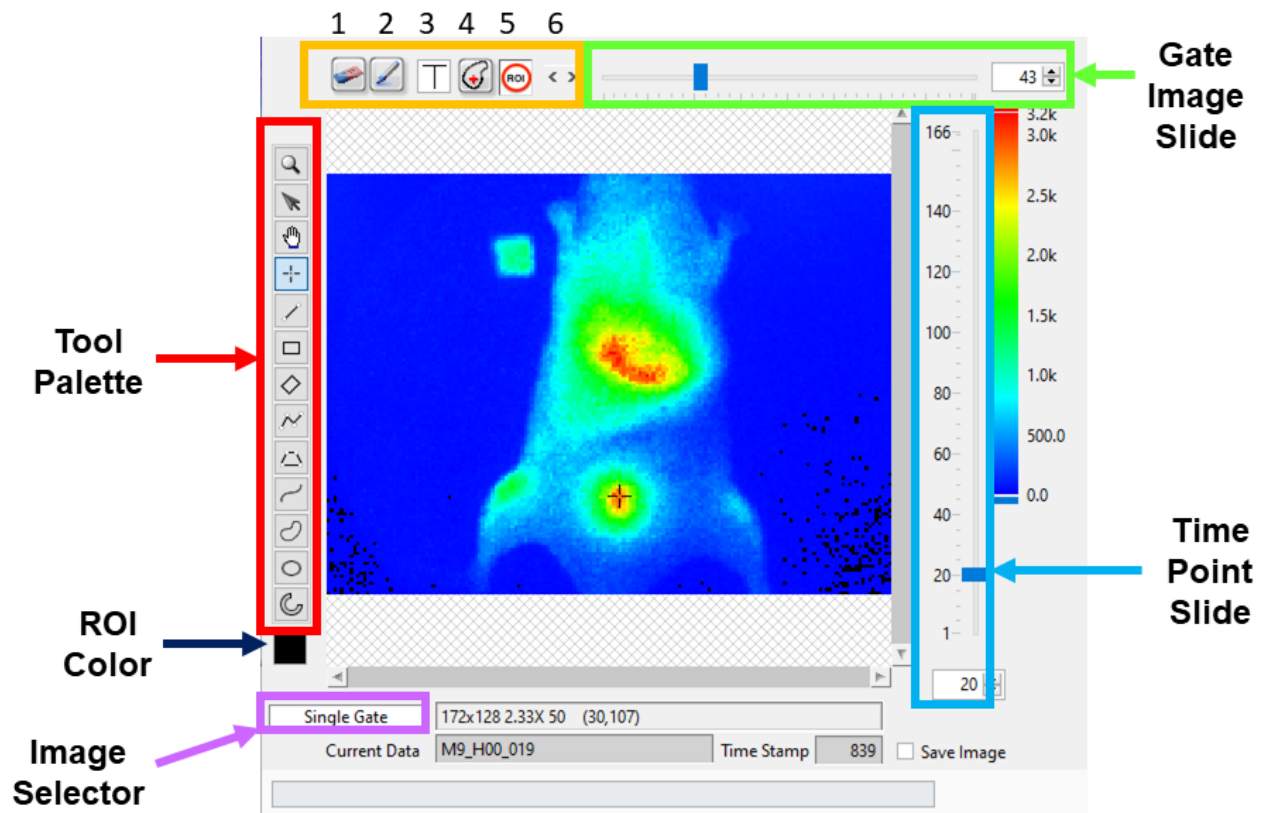
1.4 AlliGator Panels

The main window of AlliGator is divided into two main areas: the **Source Image** panel to the left, and the different analysis panels to the right.

These different panels are described individually in the following pages:

1.4.1 Source Image Panel

The *Source Image* is located on the left hand side of AlliGator's main window.



The *Source Image* can be resized by expanding the main window.

Some of the associated tools are highlighted in the snapshot shown above. Their description can be found in the following sections.

The Source Image displays different data depending on the value of the *Time Gate Slide*, on the value of the *Time Point Slide* (if it is shown) and the *Displayed Image* selector.

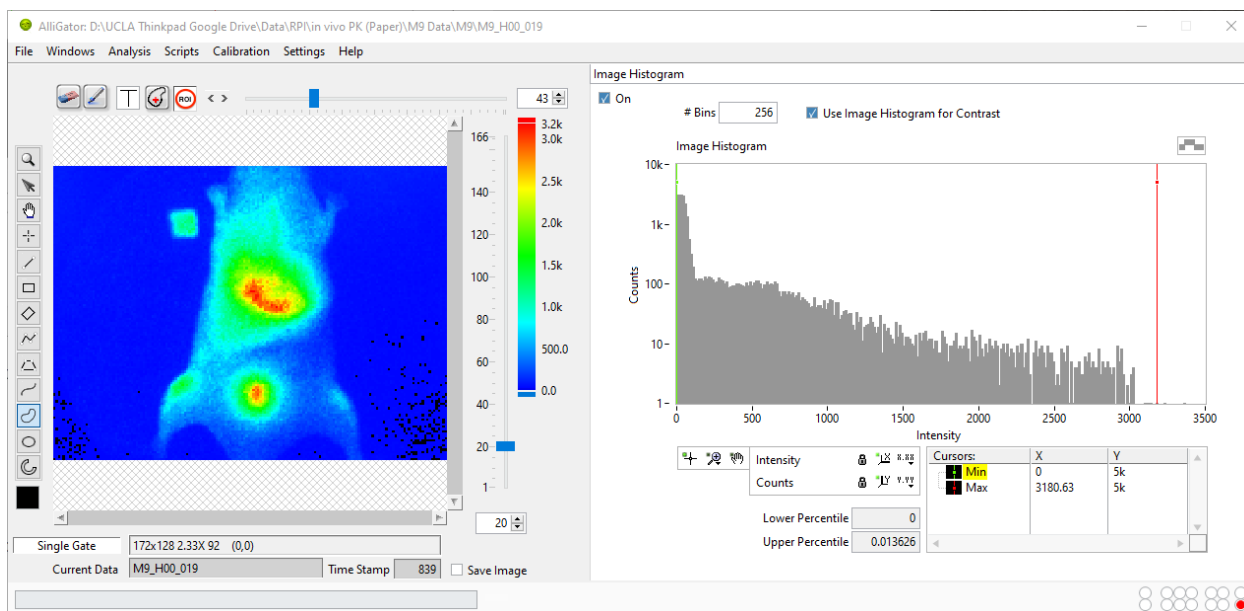
Gate Image Slide

Note: The *Gate Image Slide* is not shown when **Sum of All Gates** is selected in the *Displayed Image* selector.

Located above the image area, the *Gate Image Slide* allows browsing the individual images comprising the time gated (or binned) dataset. The numeric control on the right shows the number of gates in the dataset. Moving the slider to the left will show early gates, while late gates can be seen by moving the slider to the right. Alternatively, type in the desired gate index in the numeric control on the right.

When moving the slider, the displayed image's contrast might be inappropriate (overly dark or bright image). In this case, switch to the **Image Histogram** panel and make sure of the following:

- The *Use Image Histogram for Contrast* checkbox is checked in the **Settings:Source Image** panel.
- The *Min* and *Max* cursors are visible and encompass most of the histogram bins, as shown below:



Alternatively, uncheck the *Use Image Histogram for Contrast* checkbox in the **Settings:Source Image** panel. This will automatically adjust the contrast of each gate.

Time Point Slide

Note: The *Time Point Slide* is only visible when a Dataset Series is opened.

Located to the right of the image area, this slide allows navigating through the different time points in a time series. Move the slider to the bottom to access the early time points, or to the top to access the late time points. Alternatively, type in the desired *Time Point* index in the numeric control at the bottom.

The name of the current dataset folder is indicated in the *Current Data* text box at the bottom of the panel (in the example shown above, the folder name is M9_H00_19). If a corresponding .set file is present, the time stamp information for this dataset will be displayed next to its name (839 s in the example shown above).

While the slider is moved, the image is not updated. Image update only occurs when the slider is released, in order to reduce CPU load.

Displayed Image Selector

The *Displayed Image* selector allows switching from a mode where individual gates are shown (*Single Gate*) or only their sum (*Sum of All Gates*), or the maximum (*Max of All Gates*) or minimum (*Min of All Gates*) at each pixel across all gates. The *Sum of All Gates* mode is the default mode. If a *White Light Image* or a *ROI Mask Image* have been loaded, they can also be displayed using the selector.



Image Color Ramp

The color ramp located to the right of the source image indicates the color scale as well as the **Min** and **Max** values set in the *Image Histogram*.

The **Min** value (shown at the bottom of the scale) and **Max** value (shown at the top of the scale) define the range of pixel values which are mapped to the color scale. Pixels with value outside this range will be displayed with the **Low Color** or **High Color** shown at the bottom and top of the color scale, respectively¹.

The color scale can be changed using the *Image Source Palette* context menu. For more information on how to use and change color palettes, see the *source-image-context-menu* section below and the *Image Color Palette* page of this manual.

Image & Phasor Overlay Tools

The cluster of objects on the top-left corner of the image are used to:

1		Delete source image overlay.
2		Paint ROI points in the Phasor Plot.
3		Set the color of the painted area in the Phasor Plot.
4		Use ROI as search area.
5		Overlay mode used for the Source Image.
6		Cycle through Image ROIs.

- *Delete source image overlay*: Right-click the button to specify the button's action when pressed:
 - **All Overlays**: delete all Image Source overlays

¹ The *Low Color* box is visible as a narrow blue rectangle at the bottom of the color scale. The *High Color* box is at the top of the color section. When these colors are chosen different than the low and high colors of the color scale, pixels in the image whose values are lower or higher than the displayed range will be highlighted with the *Low Color* and *High Color* respectively. This can be useful to highlight under-exposed or over-exposed pixels, for instance.

- Source Image ROIs: delete the image ROI contours resulting from highlighting those ROIs in the Phasor Plot
- Phasor Plot ROIs Overlay: delete the overlay pixels highlighted from the Phasor Plot
- Phasor Ratio Map Overlay: delete the Phasor Ratio Map Overlay
- Phasor Plot Color Map Overlay: delete the Phasor Plot Color Map Overlay
- Decay Fit Parameter Color Map Overlay: delete the Decay Fit Parameter Color Map Overlay
- *Paint ROI points in the Phasor Plot*: highlight phasors in the Phasor Plot corresponding to the selected ROI, using the selected *Highlighted ROI Color*
- *Highlighted ROI color*: color box used to select the color used to highlight phasors in the Phasor Plot as discussed above. choosing a Transparent color (T) will result in no overlay being added.
- *Use ROI as search area*: see description in the next section
- *Overlay mode used for the Source Image*: the *Color Overlay Mode* selector allows choosing between 3 different modes of overlay from the *Phasor Plot*: ROI, Phasor Ratio/Average Lifetime Color Map and Phasor Color Map.
 - ROI: ROIs defined in the *Phasor Plot* are highlighted in the *Source Image*.
 - Phasor Ratio/Average Lifetime Map: pixels whose phasor is located in the vicinity of the segment connecting the two phasor references defined in the *Phasor Plot* are highlighted in the *Source Image* with a color reflecting their phasor ratio or average lifetime depending on the option selected in the **Settings:Phasor Plot** panel. See the phasor-ratio-map page of the manual for details.
 - Phasor Color Map: pixels are colored according to their location in the *Phasor Plot* with respect to multiple vertices defined in the **Phasor Map Color Picker** window. See the phasor-color-map page of the manual for details.

Use ROI as Search Area

The button with a red cross in it (*Use ROI as Search Area*) is used to toggle a special ROI definition mode, described next. When selected (green background), the ROI defined by the user is not the one used to compute the Fluorescence Decay and Phasor Data point of the *Phasor Graph*, but instead is used as a search area to define this ROI. The final ROI is a disk centered on the brightest pixel (in the sum image) within that search area, and *ROI Diameter* defined in the **Settings:Source Image** panel. If there are more than one pixel with the same maximum brightness, the first found from the top-left corner of the image is used.

Note that each time the Search Area ROI is moved, the button toggles back to its off position.

Image Tool Palette

The cluster of tools shown on the left side of the image are used to zoom, grab or move the image, or to draw regions of interest (ROIs). The following types of ROIs are supported (from top to bottom):

- Point
- Line
- Rectangle
- Rotated Rectangle
- Piecewise Line
- Polygon

- Freehand Curve (opened)
- Freehand Curve (closed)
- Oval
- The color box below the Tool Palette defines the overlay color in which the ROI is drawn.

Note that not all ROIs can be used for all types of analyses.

Special combinations:

- Pressing and holding down the **Alt** key while drawing a rectangle ROI will zoom on the circumscribed area (and keep a 1:1 aspect ratio)
- Holding the **Shift** key while resizing or dragging a selected ROI will modify the definition of that ROI.

Save Image

The *Save Image* checkbox at the bottom right corner of the Source Image is used to automatically save the displayed image (including overlay) each time the overlay is updated. The file is saved in the folder containing the current dataset folder, and the file name used is identical to that of the dataset folder, with an extension corresponding to the *Saved image File Format* option selected in the **Settings:Miscellaneous** panel.

If the *Displayed Image* is the **Sum of All Gates**, the file name will be followed by “Total Intensity”.

If the *Displayed Image* is a **Single Gate**, the file name will be followed by “Gate n”, where n is the gate number.

If a file with an identical name already exists, it will be overwritten.

This option is convenient for time series analysis, since it allows saving all source images in a time series (with their overlay and the palette chosen for display) in the same folder. The same function can be performed using the *Save Displayed Image with Overlay* context menu item (see below), which will open a **File Dialog** window to define the file name and location.

Context Menu

The *Source Image* context menu is shown below:

Copy Data	
Copy Image Only (with Overlay)	
Save Image Data as TIFF	
Save Displayed Image with Overlay	
Save Overlay only	
Save Image as Hot Pixel Mask	
Export Phasor Ratio Map as ASCII	
Export Average Lifetime Map as ASCII	
Palette	▶
ROIs	▶
Zoom to Fit	
Zoom to Fit (Keep Aspect Ratio)	
Source Image Plugin Example	▶

- **Copy Data**: copies the whole Source Image object bitmap image to the Clipboard for reuse in other software (or in the **Notebook**).
- **Copy Image Only (with Overlay)**: copies the visible part of the image (with overlay) to the Clipboard for reuse in other software (or in the Notebook).
- **Save Image Data as TIFF**: saves the data displayed in the Source Image as a 8- or 16-bit image, following the selected options offered to the user at the beginning of the process. The resulting file can be used for external analysis. Note that if the Sum of All Gates is shown, that sum is saved, while if a Single Gate is shown, that single gate is saved². To save a complete dataset, check the **File:Save:Dataset** submenu.
- **Save Displayed Image with Overlay**: saves the same image as copied by Copy Image Only (with Overlay)
- **Save Overlay only**: saves the current image overlay as an 8-bit image in which any non-black pixel is assigned the value 255, resulting in an effective mask image.
- **Save Image as Hot Pixel Mask**: saves the current image as a binary image (where all pixels whose values are above the max cursor are set to 1, while all others are set to 0) used for hot pixel rejection (see **Settings:Source Image**).
- **Export Phasor Ratio Map as ASCII**: when a phasor ratio analysis has been performed, exports the phasor ratio values at each pixel as a 2-dimensional matrix in an ASCII file (with File Dialog).
- **Export Phase Lifetime Map as ASCII**: when a phasor ratio analysis has been performed, exports the average lifetime computed from the phasor ratio at each pixel as a 2-dimensional matrix in an ASCII file (with File Dialog).
- **Palette**: this submenu is described in detail in the *Image Color Palette* manual page.
- **ROIs**: this submenu is described in the *Regions of Interest* manual page.
- **Zoom to Fit**: stretches the image so that it fits in the 400 x 400 pixel image display area. If the source image is not square, this will result in artificial deformations.
- **Zoom to Fit (Keep Aspect Ratio)**: as above, but ensuring that the longest dimension of the source image is stretched to 400 pixels.

Notes

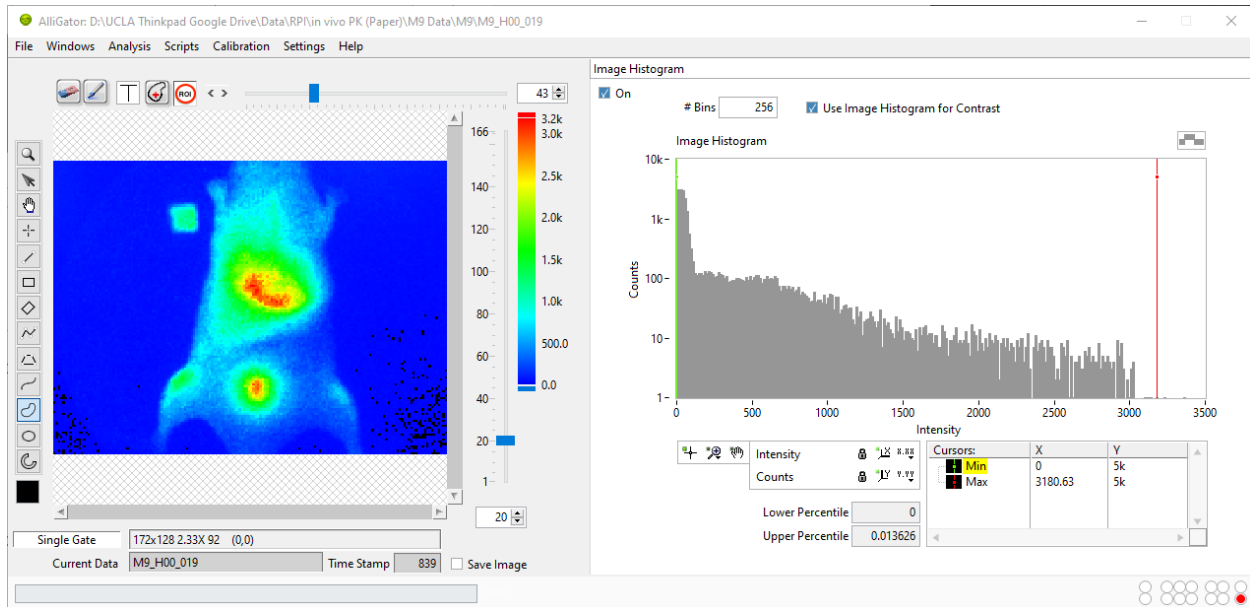
1.4.2 Image Histogram Panel

Time-gated images can contain very different numbers of counts per pixel and therefore it may be desirable to adjust their contrast either automatically or manually. Moreover, inspection of the image histogram allows defining background thresholds in order to remove background contamination from phasor plots, as discussed below.

Note that this type of processing is not necessarily useful during time series analysis and can slow it down somewhat. Therefore, it is possible to skip this step (of histogram calculation) as well as the associated image contrast adjustment step by unchecking the *On* checkbox at the top left corner of the panel.

The **Image Histogram panel** represents the pixel values of the selected gate image (when the *Single Gate* option is selected in the pull-down *Displayed Image* at the bottom left of the image) or of the sum image (if the *Sum of All Gates* option is selected) as a *n*-bin *Image Histogram*, where *n* is the value of the *# Bins* control located above the histogram:

² The options for image data export to TIFF file can be found in the **Settings:Miscellaneous** panel as *Exported Gate Settings*. See the *alligator-settings-miscellaneous* section of the manual for details.



Histogram Min & Max Cursors

Two cursors (*Min* and *Max*), defined in the bottom rightcorner of the histogram, can be used to define the image display range. They will be taken into account only if the *Use Image Histogram for Contrast* checkbox above the histogram is checked.

Any pixel in the original image with value smaller than *Min* (resp. larger than *Max*) will be represented with the color corresponding to the lowest (resp. highest) color, which can be selected individually at the bottom and top of the selected palette. The example illustrated above uses lowest and highest colors that are identical to the min and max value of the color palette, therefore no specific effect is obtained.

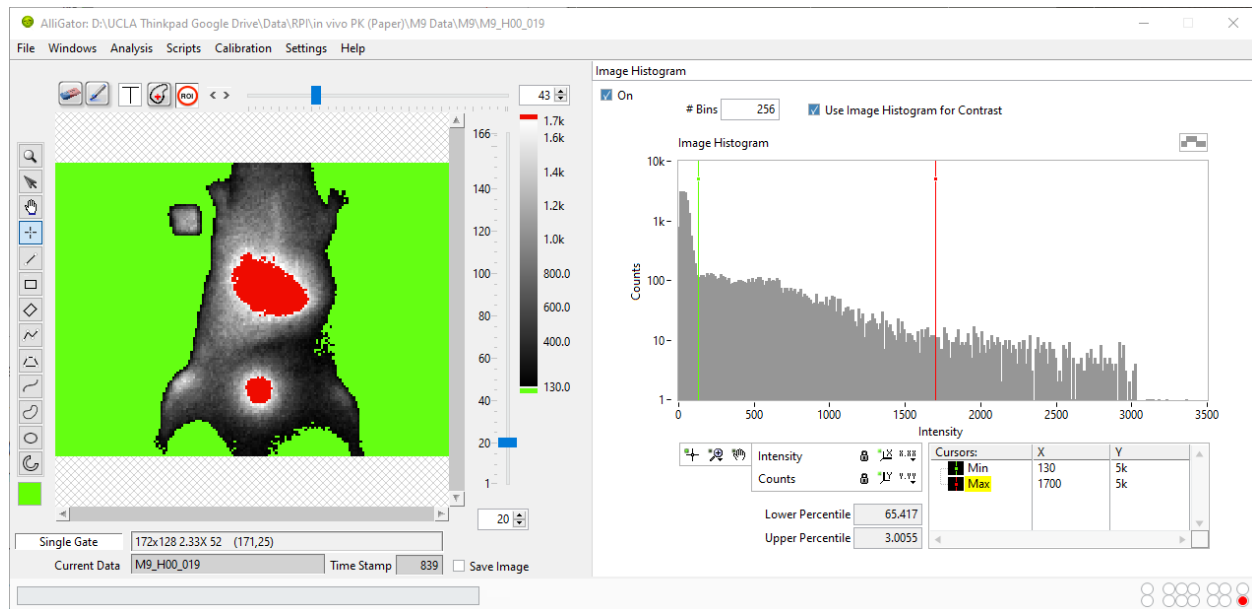
In this example, the main histogram peak to the left corresponds to background pixels, which can be set to appear green with the chosen position of the *Min* cursor (green) = 130 shown below and setting the *Lowest Color`* to be green (a different color palette -Grayscale- was selected to illustrate the effect of lowest and highest colors distinct from the palette's min and max colors)¹.

Likewise, moving the *Max* cursor below the max histogram value and choosing the highest color to be red, highlights the two brightest regions of the sample².

Note that this is only a display feature. Pixels below or above these values are still included in the phasor calculation, unless stated otherwise by the *Reject Low count Pixels* or *Reject High Count Pixels* checkboxes of the **Settings:Source Image** panel (see below).

¹ The *Lowest Color* box doesn't need to be identical to that of the *Min* cursor (and likewise for the *Highest Color* and *Max* cursor).

² The *Source Image* is updated as the user moves the histogram cursors. It is not updated if the cursors' X locations are edited in the *Cursor Legend* box, or when the lowest or highest colors are modified. In those cases, simply click in the color scale to refresh the *Source Image*.

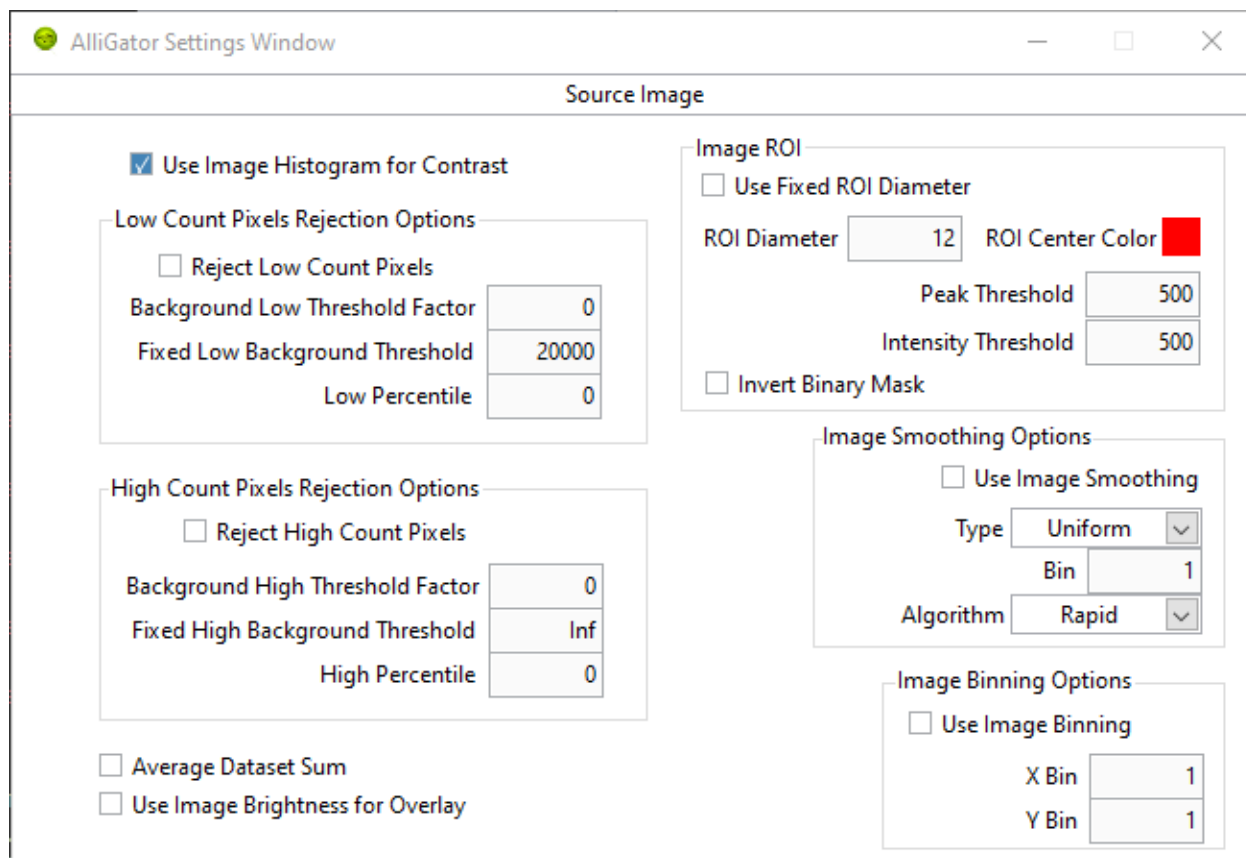


Lower & Higher Percentiles

- *Lower Percentile*: this indicator located at the bottom of the *Image Histogram*, represents the percentage of pixels whose values are smaller than the *Min* cursor location. In the image above, 65.4 % of all pixels in the image have a value smaller than 130.
- *Upper Percentile*: this indicator located at the bottom of the *Image Histogram*, represents the percentage of pixels whose values are larger than the *Max* cursor location. In the image above, 3 % of all pixels in the image have a value larger than 1,700.

Intensity Thresholds

In order to use the information provided by the image intensity histogram to instruct the software to reject low or high intensity pixels, open up the **Settings:Source Image** panel:



Two groups of controls can be used: *Low Count Pixels Rejection Options* and *High Count Pixels Rejection Options*.

1. Low Count Pixels Rejection Options

- *Reject Low Count Pixels*: if unchecked, no low-count test is performed on any pixel. If checked, the next two controls are used for each pixels.
- *Background Threshold \times Factor*: when different from zero, any pixel in the sum image of a data set, with an intensity smaller than $A \times P$, where A is the parameter's value and P is the location of the sum image histogram peak, will be ignored when computing the fluorescence decay and phasor. For instance, a value of $A = 1$ means that all pixels with value below the image histogram peak position will be ignored, while $A = 0$ means that all pixels will be retained. A nominal value to exclude background pixels appears to be $A = 3$, but it is recommended to experiment, as histogram shape can vary from sample to sample. In particular, it may happen that the histogram peak does not correspond to the most likely background value.
- *Fixed Background Threshold*: complements the previous criterion, which may fail in some particular cases. This parameter B simply sets the minimum value of a pixel in the sum image for its phasor to be computed and included in the *Phasor Plot* image, *Phasor Graph* and *Fluorescence Decay* calculations.
- *Low Percentile*: as an alternative to the two previous options, it is possible to define the minimum value of the Sum of All Gates in a pixel using this parameter LP , to be compared to the *Lower Percentile* value. A value $LP = 0$ means that no constraint is applied.

The rejection criterion uses the maximum of $A \times P$, B and the value corresponding to the LP percentile of the histogram.

2. High Count Pixels Rejection Options

- *Reject HighCount Pixels*: if unchecked, no high-count test is performed on any pixel. If checked, the next two controls are used for each pixels.
- *Peak Threshold \times Factor*: when different from zero, any pixel in the sum image of a data set, with an intensity

larger than $K \times M$, where K is the parameter's value and M is the location of the sum image maximum, will be ignored when computing the fluorescence decay and phasor. For instance, a value of $K = 0.7$ means that all pixels with value above 70% of the maximum pixel intensity will be ignored, while $K = 1$ means that all pixels will be retained.

- *Fixed Peak Threshold*: complements the previous criterion. This parameter C simply sets the maximum value of a pixel in the sum image for its phasor to be computed and included in the *Phasor Plot* image, *Phasor Graph* and *Fluorescence Decay* calculations.
- *High Percentile*: as an alternative to the two previous options, it is possible to define the maximum value of the Sum of All Gates in a pixel using this parameter HP , to be compared to the *Upper Percentile* value indicated at the bottom of the *Image Histogram*. A value $HP = 0$ means that no constraint is applied.

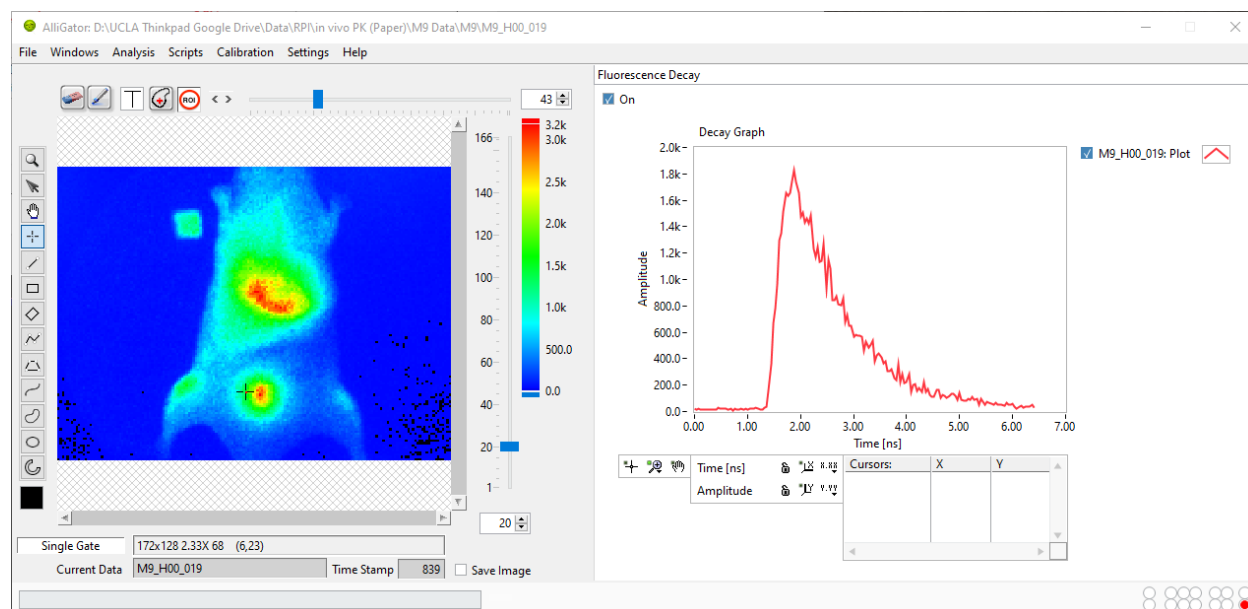
The rejection criterion uses the minimum of $K \times M$, C and the value corresponding to the HP percentile of the histogram.

Notes

1.4.3 Fluorescence Decay Panel

The **Fluorescence Decay** panel can be displayed using the pull-down list at the top-right of the main AlliGator window.

In the example shown below, a point ROI was selected in the *Tool Palette* on the lefthand side of the *Source Image*, and a ROI decay analysis performed using the **Analysis:Current ROI Analysis** menu item (Ctrl+A). While a Single Gate image is shown in the *Source Image*, the analysis encompasses all gate images, whose intensities at that selected pixels are represented as a decay curve (named **M9_H00_019: Plot¹**).



The *Decay Graph* is a feature-rich object which is comprised of different parts, some of which are common to all graph objects and are described in the [Graph Object Anatomy](#) page.

In particular, different types of contextual menus are accessible, depending on which area of the graph the user right-clicks:

- *Plot Menu*

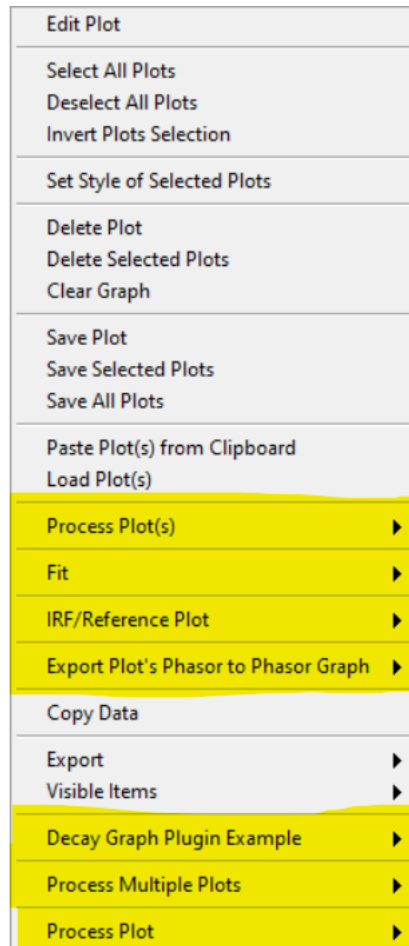
¹ When the user-selected ROI has a name (that is, when it has been stored in the alligator-source-image-roi-manager), this name is appended to the image name, instead of the generic **Plot** suffix. A number is automatically added to the decay name if the same ROI is used several times, or unnamed ROIs are used.

- *Decay Graph Custom Menu*
- *Cursor Legend Menu*

The first and last ones are similar for all graph objects and are described in detail in the *Graph Object Anatomy* page.

An overview of the *Decay Graph* custom menu is presented below, specific functionalities being described in other pages of the manual (linked to in the section below).

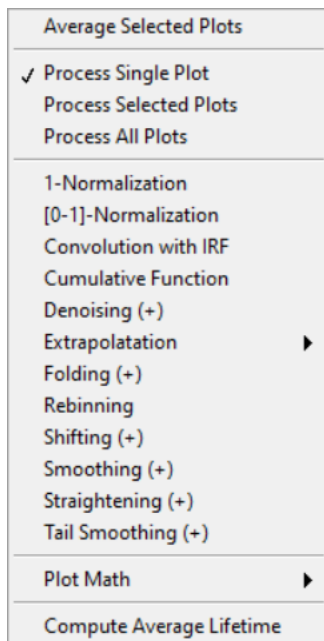
Decay Graph Custom Menu



All items not highlighted in the image above are standard graph object contextual menu items and are described in the *Graph Object Anatomy* page (*Custom Graph Menu* section).

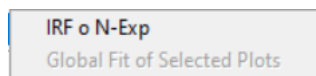
This section will briefly discuss the highlighted submenus, whose functions are described in specific pages.

- **Process Plot(s):** this menu, shown below, allows various types of plot transformations to be performed. The last category of operations are also available as *pre-processing* operations applied to each decay (before plotting in the *Decay Graph*, or other computations involving decays such as phasor calculation) and are discussed in the *Decay Preprocessing* page.

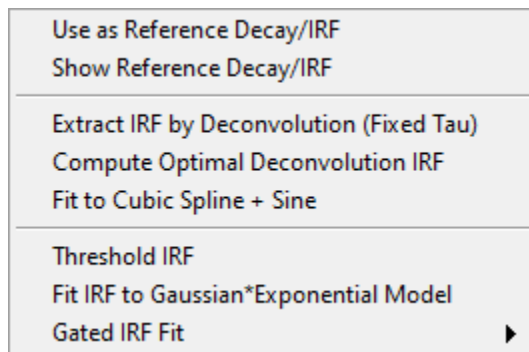


The remaining ones are:

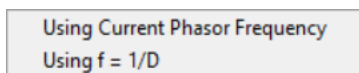
- Fit:



- IRF/Reference Plot:



- Export Decay to Phasor Graph:



- Various Python Plugins:

Python plugins menu items and submenus will appear at the bottom of the Decay Graph menu. Their names and content will depend on each installation, but typically will contain some examples or default functionalities.

Notes

1.4.4 Fluorescence Decay Statistics Panel

1.4.5 Intensity Time Trace Panel

1.4.6 Phasor Plot Panel

1.4.7 Phasor Graph Panel

1.4.8 Lifetime & Other Parameters Analysis Panel

1.4.9 Phasor Ratio Analysis Panel

1.4.10 Decay Fit Parameter Map Panel

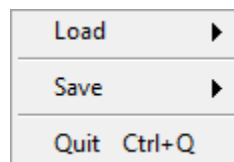
The *Decay Fit Parameter Map* panel is used to display and further process multi-ROI NLSF results (and can be used to display results from a Python plugin if needed).

The panel consist of different controls and indicators as illustrated below and discussed next.

1.5 AlliGator Menus

Some of AlliGator's functions can be performed via the top menu bar (and associated shortcuts). The menus are *File*, *Windows*, *Analysis*, *Calibration*, *Settings* and *Help*

1.5.1 File Menu



- Load

- FLI Dataset

- * Gate Image Folder: is used with data consisting of series of images
 - * HDF5 File: allows loading datasets saved in the HDF5 open source file format used by AlliGator
 - * .sdt File: loads Becker & Hickl histogrammed FLIM files
 - * .ptu File: loads PicoQuant TCSPC FLIM files
 - * .bin File: loads PicoQuant histogrammed FLIM files
 - * SwissSPAD 1 .dat File: loads a file generated by the first generation SwissSPAD camera
 - * Reload Dataset: forces reloading the current dataset (used to apply new settings)

- FLI Dataset Series

- * **Gate Image Folder Series**: is used with folders of folders of data consisting of series of images
- * **HDF5 File Series**: allows loading a folder of datasets saved in the HDF5 open source file format used by AlliGator
- * **.sdt File Series**: loads a folder of Becker & Hickl histogrammed FLIM files
- * **.ptu File Series**: loads a folder of PicoQuant FLI files
- * **.bin File Series**: loads a folder of PicoQuant histogrammed FLIM files
- **White Light Image**: opens a single image of the sample (or any other image) which will not be used for data analysis
- **Mask Image**: opens a mask image used to define regions of interest for data analysis
- **Intensity Corrections File**: opens a file containing corrections to apply to intensity values in the series
- **Save**
 - **Dataset**
 - * **Save as TIFF Gate Image Folder**: saves gate images loaded in AlliGator as TIFF files
 - * **Save as HDF5 FLI Dataset**: saves any file loaded in AlliGator in the open source file HDF5 format used by AlliGator
 - **Mask Image**: saves the current Mask Image to a TIFF file
- **Batch Conversion**: opens a window allowing the selection of multiple FLI Datasets and exporting them to HDF5 FLI Dataset files or TIFF Image Series.
- **Quit**: quits AlliGator with confirmation, offering to save the **Notebook** content (Ctrl+Q)

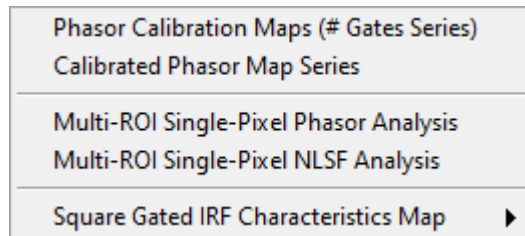
1.5.2 Windows Menu

Notebook	Ctrl+N
Dataset Information	Ctrl+I
Settings	Ctrl+E
Source Image ROI Manager	
Phasor Plot ROI Manager	
Phasor Harmonic Manager	
Image Profile	
Local Decay Graph	
Fit Status	

- **Notebook**: opens the **Notebook** window (Ctrl+N).
- **Dataset Information**: opens the **Dataset Information** window (Ctrl+I).
- **Settings**: opens the **Settings** window (Ctrl+E).
- **Source Image ROI Manager**: opens the **Image Source ROI Manager** window (Ctrl+Shift+R).
- **Phasor Plot ROI Manager**: opens the **Phasor Plot ROI Manager** window.

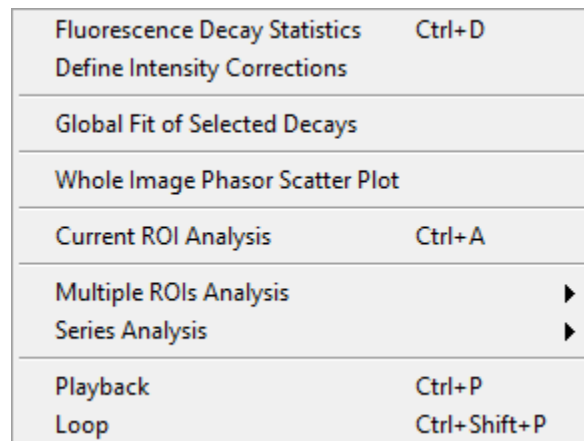
- **Phasor Harmonic Manager**: opens the **Phasor Harmonic Manager** window.
- **Image Profile**: opens the **Image Profile** window.
- **Local Decay Graph**: opens the **Local Decay Graph** window.
- **Fit Status**: opens the **Fit Status** monitoring window.

1.5.3 Scripts Menu



- **Phasor Calibration Maps (# Gates Series)**: computes phasor calibrations maps for different step parameters
- **Calibrated Phasor Map Series**: computes calibrated phasors plots for a series of dataset/gate step/calibration map
- **Multi-ROI Single-Pixel Phasor Analysis**: computes calibrated single-pixel phasors for the current ROIs
- **Multi-ROI Single-Pixel NLSF Analysis**: performs single-pixel NLSF analysis for the current ROIs
- **Square Gated IRF Characteristics Map**: fits ROI decays with one of the following gate models:
 - Square Pulse
 - Logistic Square Pulse
 - Tilted Logistic Square Pulse

1.5.4 Analysis Menu



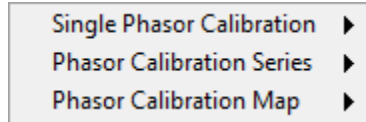
- **Fluorescence Decay Statistics**: analyzes all single pixel decays within the data set and builds an histogram of the minimum and maximum intensity in each decay. This analysis is discussed in more details in the *Fluorescence Decay Statistics* page of the manual (Ctrl+D).

- **Define Intensity Corrections:** opens the **Define & Save Intensity Corrections File** window discussed in the Intensity Corrections Definition manual page.
- **Global Fit of Selected Decays:** performs a global fit of decays selected in the Decay Graph (**Fluorescence Decay** panel). Details can be found in the ref:*Fluorescence Decay NLSF* <fluorescence-decay-nlsf> page of the manual.
- **Current ROI Analysis:** performs an analysis of the pixels located within the selected ROI or, if the Use ROI as Search Area button is pressed, in the optimal ROI within that region (Ctrl+A).
- **Multiple ROIs Analysis**
 - **All ROIs IRF Extraction**
 - * Interactive (Slow)
 - * Non-interactive (Fast)
 - * All ROIs Decay Fit
 - * Interactive (Slow)
 - * Non-Interactive (Fast, Common IRF)
 - * Non-Interactive (Fast, Individual IRF)
 - **All ROIs (Phasor) Analysis**
 - * **Interactive (Slow):** performs the same analysis as **Current ROI Analysis** on all ROIs currently defined, in the order they appear in the **ROI Manager**. In particular, it will plot the corresponding decay, add a data point to a new phasor plot in the *Phasor Graph* and display the current location of the ROI. This can take quite a long time in the case of a very long list of ROIs or for large datasets.
 - * **Non-interactive (Fast):** performs the same analysis as **Current ROI Analysis** on all ROIs currently defined, in the order they appear in the **ROI Manager**. No decay is plotted, the phasor plot resulting from the analysis is only displayed at the end, and no ROI is displayed on the * Source Image* during the analysis. This mode is recommended, as it is much faster than the previous one (> x10).
- **Series Analysis**
 - **Series Phasor:** performs phasor analysis of the loaded dataset series.
 - **Series Fit:** performs a decay fitting analysis of the loaded dataset series.
 - **Sum All Datasets in Series:** loads each dataset in a selected series and builds a dataset in which each gate image is the sum of the corresponding gate images of all datasets in the series. If the *Average Dataset Sum* checkbox in the **Settings:Source Image** panel is checked, the summed dataset is divided by the number of datasets, obtaining an average dataset.

Warning: This dataset is temporary (as indicated by the asterisk which appears after its name in the AlliGator window title). It can be saved immediately using the **File:Save HDF5 Dataset (Ctrl+S)** menu item. Any action that will update the dataset loaded in memory (such as for instance using the *Time Point Slide* next to the *Source Image*), will hide this dataset. However, it can be displayed again by selecting the **Analysis:Series Analysis:Show Dataset Series Sum** menu item.

- **Playback:** will load and display the datasets in a time series, without processing the data (contrary to the **Time-Series Analysis** menu item). It can be stopped by selecting the **Playback** menu item again or pressing the *Abort* button. (Ctrl+P).
- **Loop:** performs the same function as playback, but resumes playback from the start when the end of the series is reached. It can be stopped by selecting the **Loop** menu item or pressing the *Abort* button. (Ctrl+Shift+P).

1.5.5 Calibration Menu



The menu is comprised of three sub-menus:

- **Single Phasor Calibration**

- Use Single Phasor Calibration: performs the same action as selecting Single Phasor in the *Calibration Type* pull-down list in the **Phasor Graph** panel. Both are linked (the pull-down list is updated when the menu item is selected and vice versa).
- Save Phasor Calibration: saves the currently used single calibration phasor to a XML file.
- Load Phasor Calibration: loads a previously saved calibration phasor,
- Clear Phasor Calibration: removes the stored calibration from memory.

- **Phasor Calibration Series**

- Use Phasor Calibration Series: performs the same action as selecting Phasor Series in the *Calibration Type* pull-down list in the **Phasor Graph** panel. Both are linked (the pull-down list is updated when the menu item is selected and vice versa).
- Save Phasor Calibration Series: saves the currently used calibration phasor curve to a XML file.
- Load Phasor Calibration Series: loads a previously saved calibration phasor curve,
- Clear Phasor Calibration Series: removes the stored calibration curve from memory.

- **Phasor Calibration Map**

- Use Phasor Calibration Map: performs the same action as selecting Phasor Map in the *Calibration Type* pull-down list in the **Phasor Graph** panel. Both are linked (the pull-down list is updated when the menu item is selected and vice versa).
- Save Phasor Calibration Map: saves the currently used calibration phasor map to a XML file.
- Load Phasor Calibration Map: loads a previously saved calibration phasor map,
- Clear Phasor Calibration Map: removes the stored calibration map from memory.

1.5.6 Settings Menu

- Load Settings:

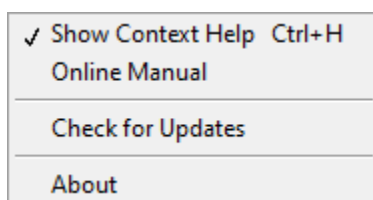
- Save Settings:

- **Export Settings to Notebook**

- All Settings: writes all settings currently used in AlliGator in the **Notebook** for future reference. Some of these settings are added to the **Notebook** during the course of data analysis. This function conveniently regroups all settings in one single location in the **Notebook**. The following items limit the content of the **Settings** window that is exported to the **Notebook**:
- Source Image
- Data Information
- Fluorescence Decay

- * All Settings
 - * Decay Pre-Processing
 - * Advanced Analysis
 - * Fit Options
 - * Fit Parameters
 - * Styles
- Settings: opens the **Settings** window

1.5.7 Help Menu



- **Show Context Help**: shows a floating window providing basic information on the object pointed at with the mouse pointer (Ctrl+H).
- **Online Manual**: opens a page in the default browser redirecting to this website.
- **Check for Update**: visits the Installation webpage to check the latest released version. If a newer version is found, the software offers to download and install it. The new version will be used on the next time AlliGator is started.
- **About AlliGator**: opens a dialog window providing basic information on AlliGator.

1.6 Alligator User Guide

The following pages introduce the user to different topics and functions of AlliGator. A first time user may benefit from reading them in the order they are listed below. Alternatively, tutorial videos found on the [AlliGator Tutorials](#) page can be consulted as either a complement or an introduction to these topics.

1.6.1 Loading & Saving FLI Datasets

The following sections detail the different ways fluorescence lifetime imaging (FLI) data can be loaded in, or saved from AlliGator.

Loading (as well as saving) functions are accessible via the **File** menu as shown in the [AlliGator Menus](#) page.

Alternatively, individual files (or file folders for time series) can be dragged-and-dropped in the main AlliGator source image panel.

The Load sub-menu is divided into three sub-menus: FLI Dataset, FLI Dataset Series, and Other Files.

The Save menu only support saving a single dataset in either of two formats: Image series, or HDF5 FLI dataset.

Dataset Series are discussed in more details in the time-series-analysis pages of the manual.

Introduction

AlliGator was initially developed to analyze data from time-gated ICCD cameras. Since this type of data is similar to that of FLI performed with time-correlated single-photon counting (TCSPC) hardware, support was added for this kind of data as well (both time-stamped and binned data are supported).

Data in time-gated or binned decay analysis generally consists in a sequence of G images.

Each image in the sequence represents a “gate” image. A gate is an image acquired with the image intensifier (or more generally detector) rapidly turned on at a specific time after the laser pulse (gate offset t_i , where i is the gate image index in the sequence) and rapidly turned off after a constant duration (gate duration or width W). This process is repeated over many laser periods and the data accumulated in the gate image. Subsequently, the delay of the on/off gating process is changed, and a new gate image is acquired. In this process, each gate is separated from the next by a constant step (or gate separation dt).

Time-stamped TCSPC data consists by contrast in individual photon information (position (X, Y) in the image, nanotime t representing the arrival time with respect to the laser (generally an additional macrotime is provided, which will not be discussed here for the sake of simplicity). This type of data is generally transformed into binned data by collecting all photons in each pixels and histogramming their arrival times in a predetermined number G of adjacent bins (of duration W equal to their separation dt). This data in turn can be looked at as a sequence of “binned decay” images, which are similar to the gate images discussed previously in the case of gated acquisition.

Although each gate image takes some time to acquire, and a sequence of images takes about G times this amount, we will refer to such a sequence of images as a “time point”.

AlliGator allows the analysis of individual time points, or a series of such time points (i.e. a time series). Loading a single time-point or a time-series is done differently as described next.

single HDF5 FLI Dataset

A simple open source file format in which to save a variety of different files from different sources was introduced with version 0.16 of AlliGator. It simplifies data storage (using a single file instead of a folder of images) and loading (for instance, in the case of SwissSPAD 2 data, pre-processing of raw data is not necessary anymore, once saved as an AlliGator HDF5 file). In addition, this format supports floating point values for gate image pixel intensity, which allows saving processed datasets without loss (e.g. background-subtracted or pile-up corrected datasets will be comprised of non-integer pixel values). Finally, the format includes a lot of metadata which helps with traceability and reproducibility.

Details about the format itself can be found in the [AlliGator HDF5 File Format](#) page of this manual.

To load an AlliGator HDF5 file, use **File:Load:FLI Dataset:HDF5 File** (Ctrl+O). The path to the dataset will be displayed in the title bar. Alternatively, drag and drop the file in the Source Image panel.

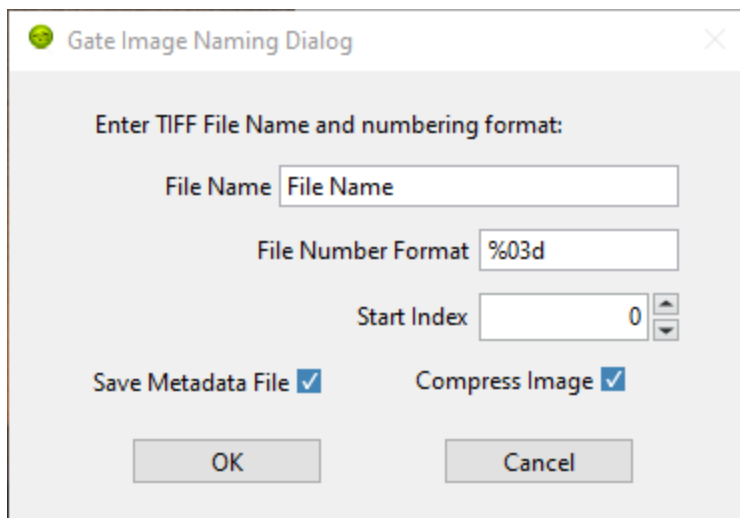
To save a dataset (irrespective of its source), use **File:Save:Dataset:Save as HDF5 FLI Dataset** (Ctrl+Shift+S).

Folder of Gate Images

To load a single time point (consisting of G gate images), use **File:Load:FLI Dataset:Gate Image Folder** (Ctrl+L). The path to the dataset folder will be displayed in the title bar. Alternatively, drag and drop the folder in the Source Image panel.

Supported gate image file formats are: BMP, TIFF, JPEG, JPEG2000, PNG. The files can be 8 or 16 bits gray scale images.

To save a FLI dataset as a series of gate images, use **File:Save:Dataset:Save as TIFF Gate Image Folder**. This will first open a **Gate Image Naming Dialog** window where the user can define the name (prefix) of individual gate images, as well as define additional parameters:



Next, a file dialog window allows selecting where to save the gate images.

Note that no additional information is saved, therefore it is recommended to include additional information needed to reload (or at least make sense of) these images in an auxiliary readable file.

Becker & Hickl .sdt FLI Dataset

To load an histogrammed .sdt file saved by a Becker & Hickl FLIM electronics, use **File:Load:FLI Dataset:.sdt File**. The path to the dataset will be displayed in the title bar.

PicoQuant .ptu Dataset

PicoQuant FLIM electronics can save data as individual photon time stamps with spatial information (.ptu files) or as histogrammed data (.bin files).

To load a .ptu file, use **File:Load:FLI Dataset:.ptu File**. The path to the dataset will be displayed in the title bar. Note that the user needs to specify how to interpret the photon time stamps by providing a number of bins *G* in which to sort out the photons via the **# Gates** parameter defined in the **Settings:Data Information** panel⁰.

PicoQuant .bin Dataset

To load a .bin file, use **File:Load:FLI Dataset:.bin File**. The path to the dataset will be displayed in the title bar.

⁰ It is recommended to check that decays computed with the user-specified **# Gates** parameter do not suffer from binning artefacts. If that is the case, adjust the number of bins (i.e. **# Gates**) until these artefacts disappear.

Reloading a Dataset

To update a dataset after modifying an option requiring reloading the dataset to take effect (such as for instance the number of gates), use **File:FLI Dataset:Reload Dataset (Ctrl+R)**

1.6.2 Loading & Saving FLI Dataset Series

Folder of HDF5, .sdt or .ptu Datasets

In order to load a time series (or any succession of datasets to be analyzed as a series) consisting of individual FLI datasets of a single kind (.hdf5, .sdt, .bin or .ptu), make sure that they are grouped in a single folder. This folder can contain other file types, which will be ignored when loading the series.

In order to load a time series (or any succession of datasets to be analyzed as a series) consisting of gate images, use **File:Load:FLI Dataset Series:xxx File Series**, where xxx stands for .hdf5 or .sdt or .bin or .ptu. The HDF5 File Series loading option can be invoked with the Ctrl+Shift+O keyboard shortcut.

Folder of Folders of Gate Images

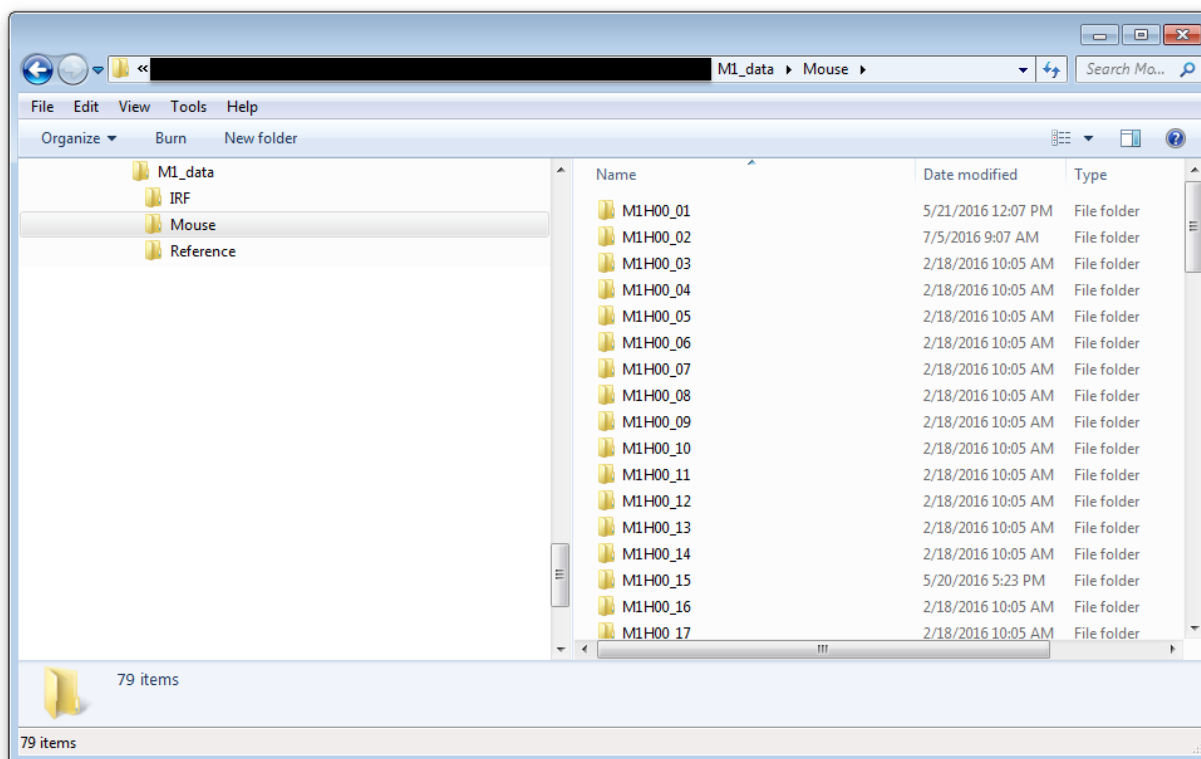
In order to load a time series (or any succession of datasets to be analyzed as a series) consisting of gate images, use **File:Load:FLI Dataset Series:Gate Image Folder Series (Ctrl+Shift+L)**. In the case of LaVision ICCD data, it is possible to use the time stamp of each dataset saved in the associated .set files. To enable this, check the *Use File Timestamp* checkbox in the **Time Trace** panel of either the **Settings** or **AlliGator** windows, before loading the time series.

After the folder containing the time series has been selected, the first data set in the series will be loaded and displayed in the Source Image indicator as described earlier.

In addition, a vertical slide (*Time Point Slide*) will be displayed on the right-hand side of the image, allowing to explore the time series. The name of the data set currently displayed will be indicated in the *Current Data* text box below the image.

Note that to avoid slowing down the software, there is no update of the image as the vertical slide is moved around: only the name of the *Current Data* is updated. As soon as the slide is released, the corresponding data set is loaded. Occasionally, the software may lose track of the slide being moved. Click on it or enter the desired dataset index in the associated control to force an update.

Each time point is a folder identified by a name specifying its order in the time series. In other words, a time series with P time points will look something like this on disk:



or, more generally:

time series/time point 1/image 1 time series/time point 1/image 2 ... time series/time point 1/image N

time series/time point 2/image 1 time series/time point 2/image 2 ... time series/time point 2/image G ...

time series/time point P/image 1 time series/time point P/image 2 ... time series/time point P/image G

time series is the name of the folder (Mouse in the figure above) in which all time point subfolders are located (M1H00_nn in the figure above). These subfolders should be named using a common root name followed by an increasing number suffix. For instance, folders named TimePoint_001, TimePoint_002.tif, ..., TimePoint_100.tif constitute a valid series of names, but TimePoint1, TimePoint2, ..., TimePoint10,... etc. is also an appropriate naming convention⁰.

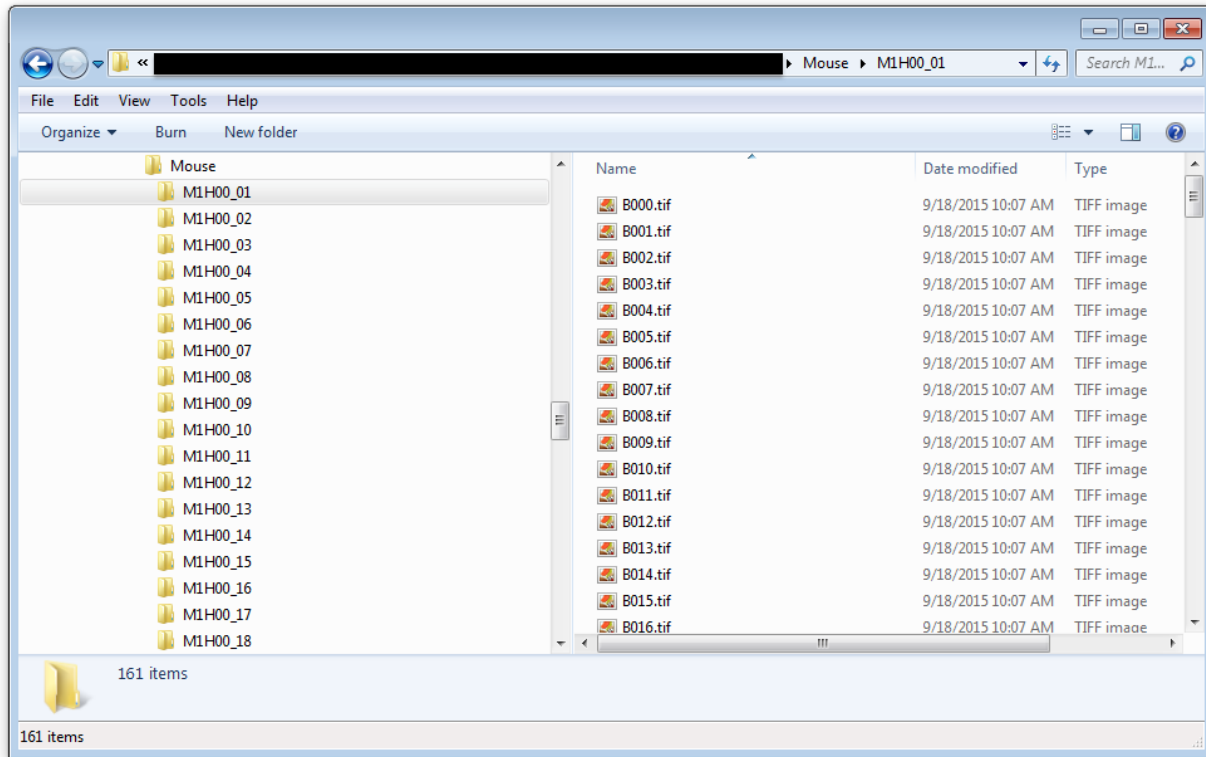
The naming convention for images in each folder should follow a similar pattern⁰: root name followed by a numeric suffix. The software will assume that the files, ordered numerically (using their suffix) are also ordered temporally, i.e. correspond to successive gates, starting at offset 0 and incremented by a constant step equal to the specified *Gate Separation* parameter (see the [::ref::fluorescence-decay-panel](#) page of the manual).

For instance, files named Image000.tif, Image001.tif, ..., Image100.tif constitute a valid series of names, but other naming conventions can be used. For instance, Image1.tif, Image2.tif, ..., Image10.tif,... etc., is also an appropriate naming convention.

An example of image folder is shown below:

⁰ The LaVision acquisition software seems to name folders in a time series in the following manner:

⁰ The LaVision camera recently (2017) decided to output gate files with names of the kind: root_nameXXXXX_Delay=YYYps.tif, where XXXXX is the gate number and YYY is the gate delay with respect to the laser trigger. Versions of AlliGator above 0.9.13 support this unconventional file name format.



Notes

Folder Folder_1 Folder_2 etc.

This unfortunately is not compatible with the algorithm used to figure out the common root name of all folders as well as their order. Fortunately, the fix is simple and consists in renaming the folder corresponding to time point 0 (Folder in the example above) as Folder_0.

1.6.3 Regions of Interest

Regions of interest (ROIs) are defined by a closed contour and can be created in different ways:

- by drawing them using the drawing tools (to the left of the image)
- by loading a ROI Mask image (this only works for the Source Image)
- by pasting a text ROI description
- using the Define ROI(s) above Min Peak context menu of the Image Source

These approaches are described below.

Drawing a ROI

There are a few default shape categories available on the left hand side of the image to draw ROIs:

- Rectangle
- Tilted Rectangle
- Polygon (double-click the last point to finish and close the shape)
- Free Form
- Oval
- Annulus

To draw a ROI comprising several shapes, hold the **Ctrl** key down while drawing the successive shapes.

ROIs are represented as a special kind of overlay on top of the image and can be manipulated using the **Selection** tool:



This tool acts as either a resizing tool or as a moving tool.

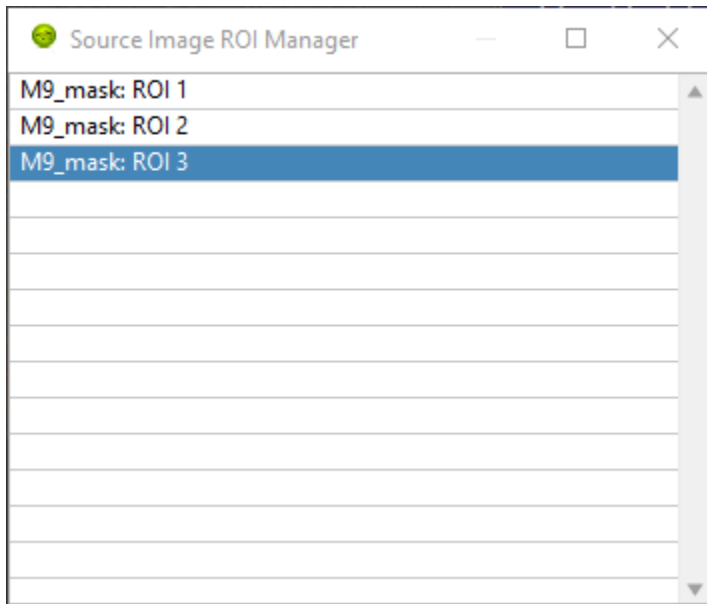
Managing ROIs

ROIs can be managed using the ROIs context menu functions:

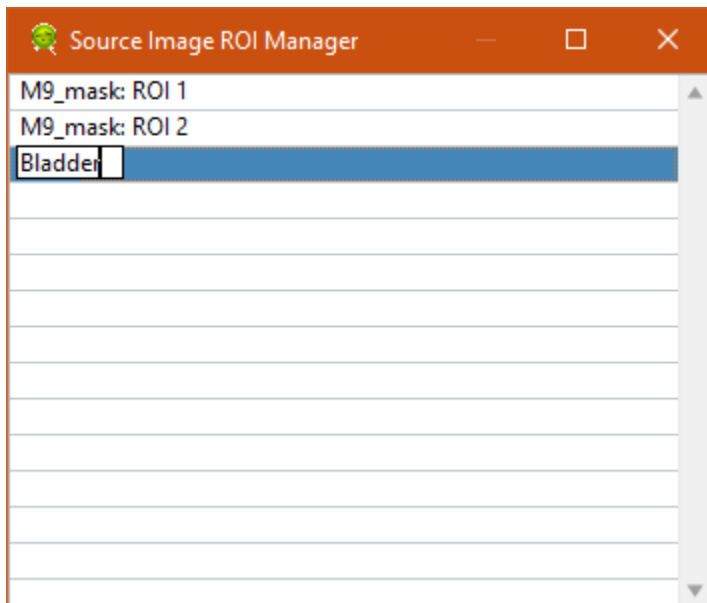
Source Image ROI Manager
Next ROI Previous ROI
Show All ROIs Hide All ROIs
Merge All ROIs
Store ROI Delete ROI Delete All ROIs
Save Current ROI Save All ROIs Load ROI(s)
Paste ROI Definition
Create ROI(s) from Pixels with Peak above Min Create ROI(s) from Pixels with Intensity above Min Create ROI Grid ROI to Individual Pixel ROIs
ROI Statistics Export ROI Description to Notebook

Source ROI Manager

The ROI manager is a floating window used to select stored ROIs by name:

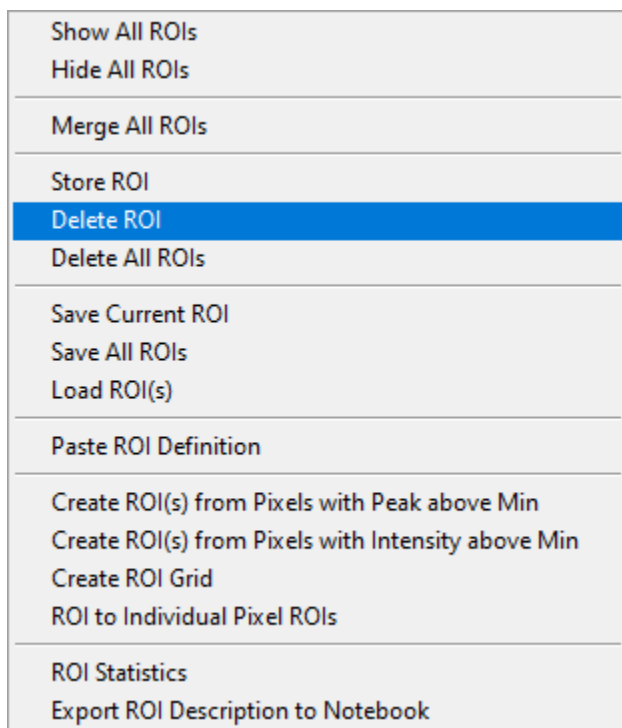


ROI names can be modified by clicking on an already selected ROI name, making it editable (when the background turns black):



If the modified name is already used by another ROI in the list, a number is automatically appended to the name.

It is also possible to delete or save ROIs from within the ROI manager using the right-click menu (see ROI Saving/Loading below).



It is also possible to navigate through the different stored ROIs using the navigation buttons described next.

ROI Navigation

In addition to the ROI Manager window, it is possible to cycle through the stored ROIs using the two little arrow heads/navigation buttons located to the right of the **Overlay Mode** button above the **Source Image**:



Alternatively, use **Next ROI** or **Previous ROI** in the right-click menu of the **Source Image ROI Manager** to the same effect.

ROI Display

To hide the visible ROI(s), select **Hide All ROIs**. Contrary to the **Delete All ROIs** function discussed next, this does not destroy the stored ROI(s).

ROI(s) can be displayed again by either using the ROI Navigation buttons or by selecting a ROI in the ROI Manager window.

To show all stored ROIs, select **Show All ROIs**. Note that this does not select any new or different ROI.

ROI Storage

ROI Storage

A ROI can be used right after it has been defined, but any click elsewhere in the image with a ROI-modifying tool (selection or drawing tool) will erase it.

To avoid this, you can store the ROI after it has been defined by using the **Store ROI** right-click menu item.

To remove a ROI from memory, cycle through the ROIs to show it and use **Delete ROI**.

To remove all ROIs from memory, use **Delete All ROIs**.

Saving and Loading ROIs

To save a ROI for future use, it is possible to save it as a file using **ROIs->Save Current ROI**. Use a self-explanatory name to simplify the task of finding the appropriate ROI next time you need it. The file extension is automatically set to **.roi**.

It is also possible to save all ROIs in a single file, using **Save All ROIs**. The corresponding file extension is automatically set to **.roiN**.

To load a saved ROI (or multiple ROIs saved together), use **Load ROI**.

It is also possible to define ROIs using a binary image as described in a later section.

Note: Prior to version 0.17, a XML file format was used, which turned out to be a bit verbose and slow to load. From 0.17 onward, the file format was changed to JSON to mitigate these issues. Older XML files can still be loaded, but it is recommended to save them (e.g. overwriting the older version) in the new format.

Importing ROI description

Another method to create a ROI is to copy/paste a text description of a ROI generated by AlliGator and output to the **Notebook** during analysis. This is particularly convenient if a user has saved the Notebook after data analysis, but either failed to store and save the different ROIs used, or the corresponding files are not available.

To recreate a ROI from its description, simply select it in and copy it from the Notebook (or another text editor), and moving to the **ROI Manager**, right-click and select **Paste ROI Definition**. If the definition is valid, this will create a new ROI and add it to the ROI Manager's list.

For instance, here is a ROI definition representing an ellipse:

```
Number of pixels: 2728
Global Rectangle (LT, BR): (234, 211), (278, 273)
Number of Contours: 1
Contour 1: Oval
Points: (234, 211), (278, 273)
```

ROI Definition

Defining ROI(s) by a decay peak threshold

The **Source Image** panel of the **Settings** window has a parameter named **Peak Threshold**. This parameter relates to the maximum value of the single pixel decay (peak value).

It is possible to define ROI(s) based on pixels whose peak value is larger than this **Peak Threshold** by using the right-click menu **Create ROI(s) from Pixels with Peak above Min**.

[The appropriate value to use can be estimated by an analysis of the **Fluorescence Decay Statistics** histograms computed in the corresponding panel (see the corresponding manual page: [Fluorescence Decay Statistics Panel](#))].

The result of this operation is one or more ROIs meeting this criterion. The ROIs are stored and added to the pool of existing ROIs.

Defining ROI(s) by a image intensity threshold

The **Source Image** panel of the **Settings** window has a parameter named **Intensity Threshold**.

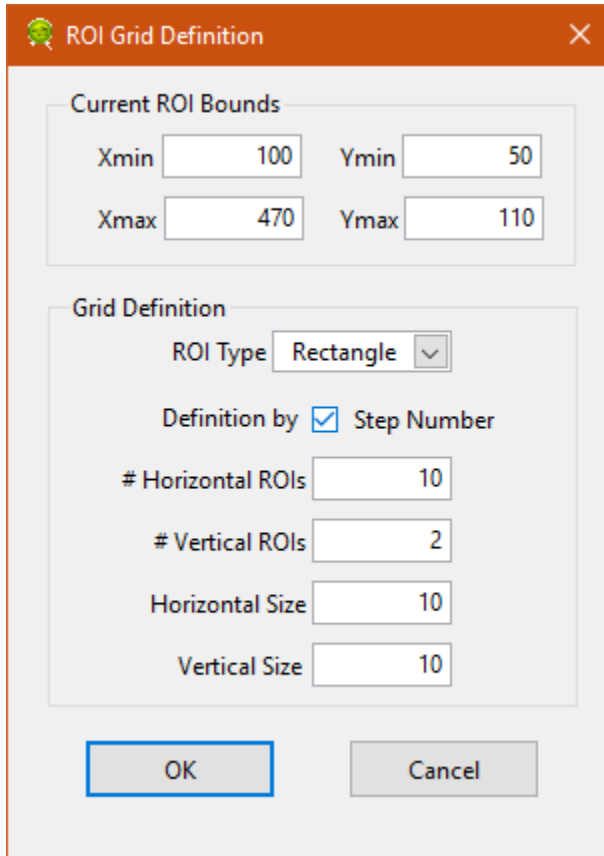
It is possible to define ROI(s) based on pixels whose total intensity is larger than this **Intensity Threshold** by using the right-click menu **Create ROI(s) from Pixels with Intensity above Min**.

[The appropriate value to use can be estimated by an analysis of the **Image Histogram** computed in the corresponding panel.

The result of this operation is one or more ROIs meeting this criterion. The ROIs are stored and added to the pool of existing ROIs.

ROI Grid

This menu item opens up a dialog allowing the definition of a series of rectangular ROIs arranged in a grid pattern:



ROI Grid Definition

Current ROI Bounds

Xmin: 100 Ymin: 50

Xmax: 470 Ymax: 110

Grid Definition

ROI Type: Rectangle

Definition by: ☒ Step Number

Horizontal ROIs: 10

Vertical ROIs: 2

Horizontal Size: 10

Vertical Size: 10

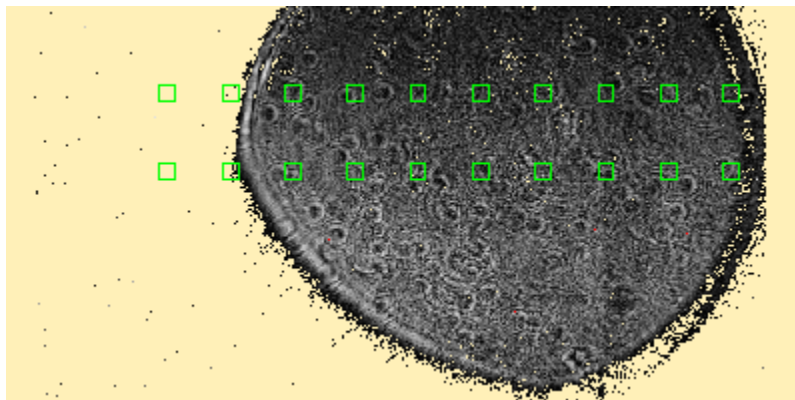
OK Cancel

If a ROI is displayed in the **Source Image**, its bounding box definition will be used as **Current ROI Bounds** parameters. Otherwise, define the bounding rectangle for the grid:

(Xmin, Ymin) = coordinate of top-left corner (Xmax, Ymax) = coordinates of bottom-right corner

The **Grid Definition** cluster of parameters is used to define the type, number and size of the individual ROIs arranged on the grid. The grid can be defined either in terms of **Step Size** or **Step Number**, depending on the **Definition by** checkbox.

Using the definition shown above results in the following set of ROIs:



The corresponding ROIs are added to the ROI list (visible in the **ROI Manager** window), with names of the kind: **Type ROI (X,Y)+(W,H)**, where Type is the ROI type (Point, Rectangle or Oval), (X,Y) are the coordinates of the top-left corner of the ROI and (W,H) are the width and height of the ROI.

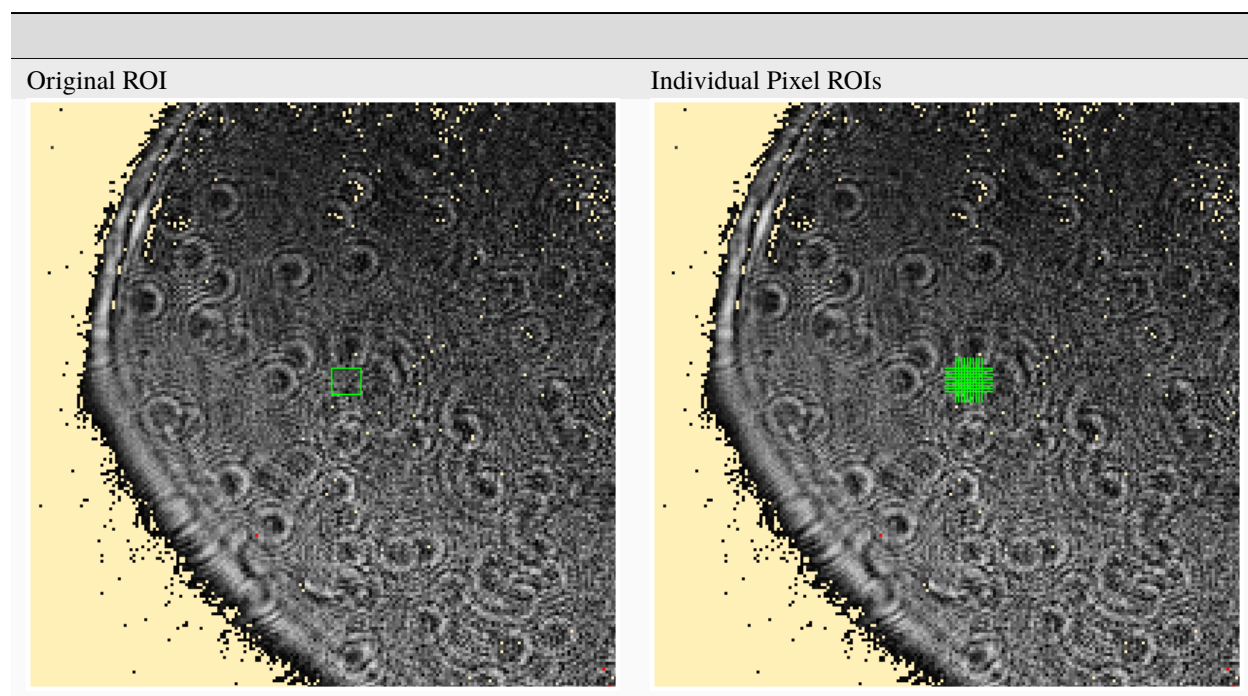
Note: To see all ROIs as in the figure above, use **Show All ROIs** in the **Source Image** context menu.

Convert ROI to Individual Pixel ROIs

To obtain all individual pixel decays within a ROI, it is possible to create “single pixel” ROIs for all pixels within that ROI with the ROI to Individual Pixel ROIs function.

The individual pixel ROIs are added to the ROI list (visible in the **ROI Manager** window), with names of the kind: Source ROI (X,Y), where Source ROI is the name of the original ROI (or Temp ROI if the original ROI was not stored) and (X,Y) are the pixel coordinates.

Note: To see all ROIs as in the figure above, use Show All ROIs in the Source Image context menu:



Warning: Depending on the PC used, it might be recommended to use this feature only with ROIs of moderate size, due to the potential time and memory consumption of this operation. In particular, it is not recommended to select the whole image and create individual pixel ROIs for all pixels for large images.

ROI Information

ROI Statistics

This menu item computes basic image statistics for the pixels of the displayed image contained in the ROI:

Area (pixels)
 % of Image
 Total Intensity
 Mean Value
 Standard Variation
 Min Value
 Max Value

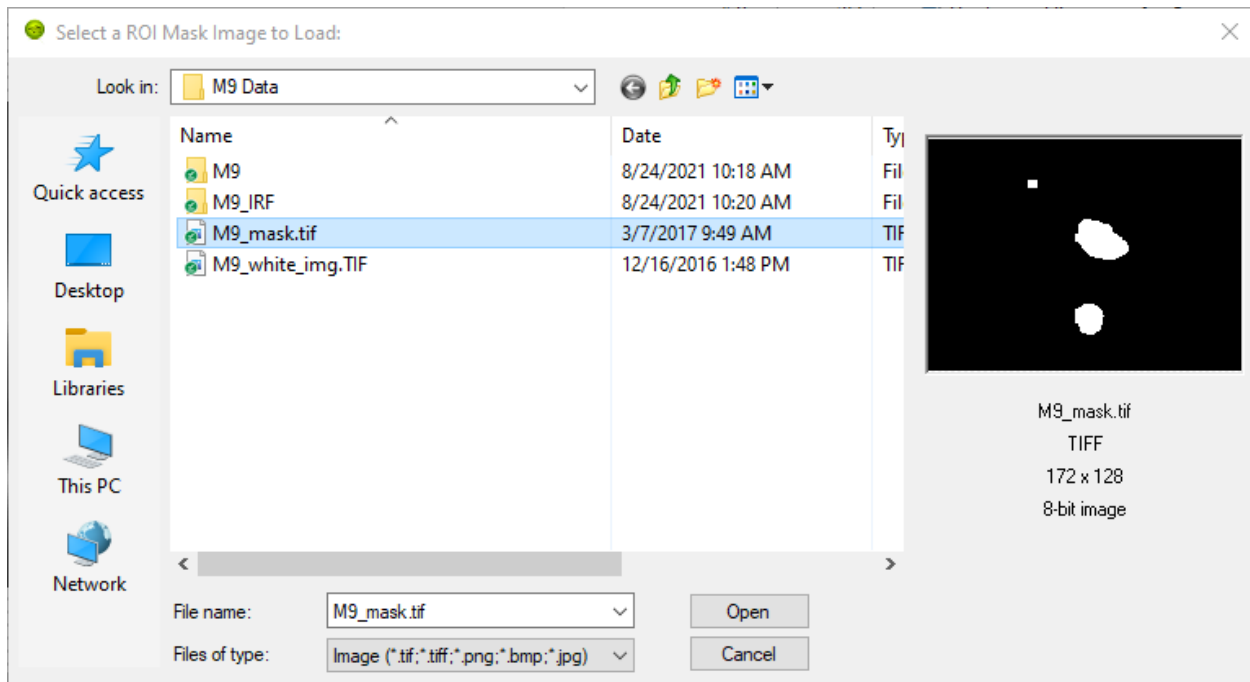
The quantities are self-explanatory, except maybe for the % of Image quantity, which is simply the area of the ROI divided by the total image area.

Export ROI Description to Notebook

While exporting a ROI description as a file is the best way to preserve information about a ROI definition for later re-use within AlliGator, it is also possible to export a more readable description of it to the Notebook using the **Export ROI Description to Notebook** menu item.

Using a ROI Mask Image

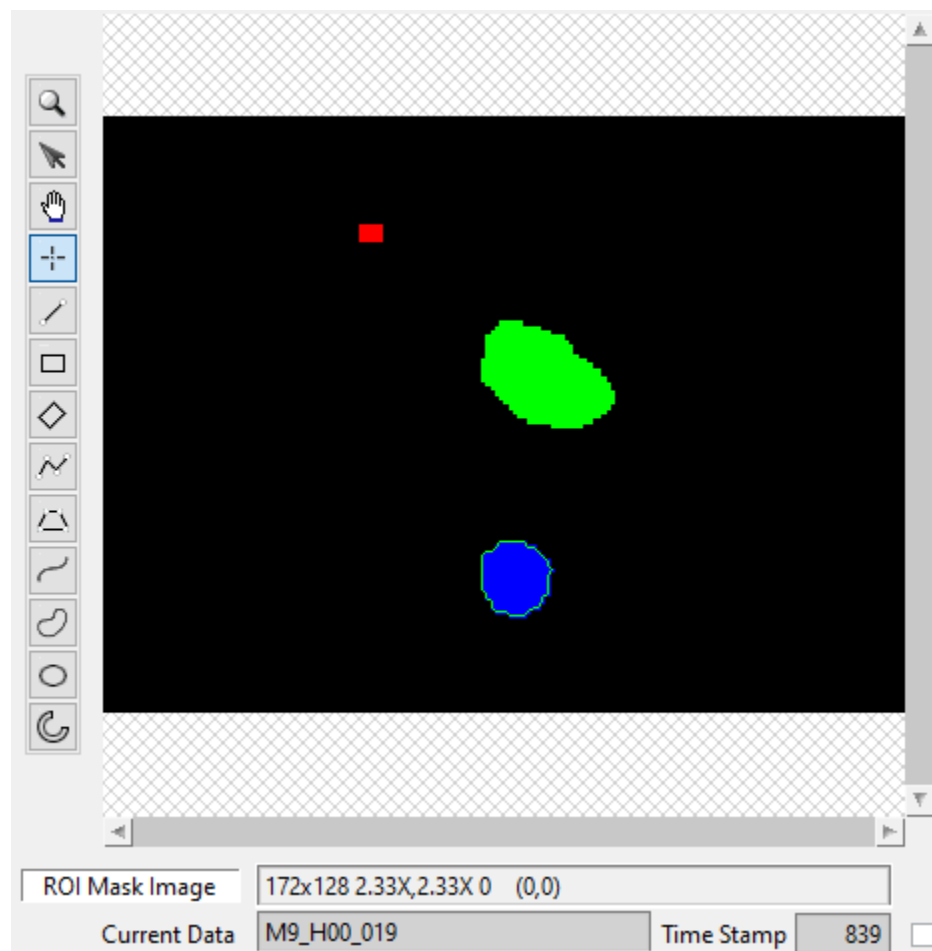
Using ROIs defined externally is possible by providing an 8-bit ROI Mask Image in which background pixels have a value of 0 and pixels to include in the ROI(s) have a value of 255. Such an image needs to be loaded using the **File:Load:Other Files:Open ROI Mask** menu item, as discussed in the [File Menu](#) section of the manual:



ROIs are extracted from this image using a “particle” detection algorithm which groups contiguous pixels with similar grayscale values into separate domains (or particles). In the example shown above, the mask image is comprised of two types of pixel values: 0 (black) or 255 (white). The algorithm will identify 3 groups (in the above example) and associate a ROI for each one of them. It will also “paint” each region with a different grayscale value (starting with 1 for the first particle, 2 for the second, etc.).

To invert the image before processing it, check the **Invert Binary Mask** checkbox in the **Source Image** panel of the **Settings** window.

The resulting processed image will be displayed in the **Source Image** control using a binary color scale, which associates a very different vivid color with each consecutive pixel value.



Notice how the **Displayed Image** pull-down list reads ROI Mask Image. It is possible to visualize this image at any time by selecting this value.

When loading the mask image, AlliGator will identify the different ROIs and provide them with default name constructed from the file name and the ROI order.

These names can later be modified using the **ROI Manager** window.

Internally, AlliGator stores one definition (the whole contour of each ROI) per ROI, even when these ROIs are loaded from a single .roiN file. These definitions can be cycled through using the ROI Navigation buttons described above or the **ROI Manager** window.

1.6.4 Decay Pre-processing

A number of options and operations can be applied to ROI decays. Those are defined in the **Settings:Fluorescence Decay:Basic Analysis** window.

Warning: Some options only take effect after the file is reloaded. Those are indicated with an asterisk in parenthesis (*) after their label in the Settings window, and in the text below.

Up until version 0.18.1, the order of operations was:

- Pile-up Correction
- Background Correction
- Revert

- Normalize
- Smoothen
- Shift
- Extrapolate
- Straighten

In version 0.18.2, the order of operations was changed to:

- Pile-up Correction
- Background Correction
- Revert
- Smoothen
- Straighten
- Shift/Rotate
- Extrapolate
- Normalize

Moreover, the way the “Shift” operation functions was modified in 0.18.2, to include an option to apply a constant shift (the “Default” mode) and an option to make this shift “Rotate” the decay, by assuming that the dataset is periodic.

In later versions (0.19 and following), the order of the decay pre-processing operations (except for pile-up correction, which is always performed first) became user-selectable. In the case of photobleaching or photobrightening, this allows correcting for that effect before attempting a square-gated single-exponential background subtraction, for instance. To change the order of operations, right-click on the Operations Order list and select Reorder operations: This will open a window showing the current list order: Selecting any of the items, use the buttons at the bottom to change the location of that item in the list. When done, click the OK button (rightmost button) to accept and close the window, or the Cancel button (leftmost button) to cancel any modifications to the original list. Closing the window is equivalent to cancelling and accepting that choice.

1. Gate Selection

1.1 Gate Duration: The gate duration parameter is currently only used to calculate the Universal Curve (aka pseudo universal circle).

1.2. Gate Separation: The gate separation specifies the temporal offset between gates.

1.3. Gate Step (*): The default calculation mode is to use all gates to build the decay plot. However, it is possible to use only one every n gates, by specifying n in the value of Gate Step. The effect of choosing to use $n = 8$ (rather than the default $n = 1$) is shown on the figure below. 1.4. Gates to Skip (*): Two other parameters defined in the Settings>>Fluorescence Decay>>Basic Analysis panel affect the final shape of the decay:

Decay Points to Skip: Start Decay Points to Skip: End

These parameters define how many gate images are omitted when loading the datasets. Specifically, if Start = s , End = e , and each dataset is comprised of N gates, only gates $s+1, s+2, \dots, n - e - 1, n - e$ will be retained in the analysis. This can be useful if decay inspection reveals some unphysical “feature” at the beginning or end of the decays.

Warning: Changing these two parameters after the dataset has been loaded has no effect on subsequent analysis. For these parameters to take effect, the dataset needs to be reloaded.

2. Pile-up Correction (*)

When selected, this option gives access to the pixel well capacity (Max Value), which, in SwissSPAD data, corresponds to the number of 1-bit frames accumulated for each gate image. The correction applied takes into account the possibility of pile-up (missed counts) at high count rates, according to:

$$S = -N \log(1 - R/N)$$

where R is the raw count, N is the pixel well capacity parameter and S the corrected count value.

3. Background Subtraction

The Background Subtraction checkbox is used to perform automatic background subtraction from all computed decays (as well as when computing the Phasor Plot Image) in one of the following manners.

3.1. Baseline subtraction based on a reference window

As can be noticed in the plot shown above, it is customary in time-gated imaging experiments to offset the decay on the time axis in such a way that a flat (background) zone precedes the actual rise and decay. This can therefore be used to estimate the background for each ROI (and by extension, for each pixel). In AlliGator, the user needs to define the Min Gate and Max Gate values to be used for background estimation. This is done in the Settings>>Fluorescence Decay panel (see Figure below). The actual value retained as background is the minimum (Min) or average (Mean) value recorded in this gate interval, depending on the chosen Method.

3.2. Square Gated Single-Exponential method

In this approach (Square Gated Single-Exponential), the recorded decay is assumed to be the result of a single-exponential decay integrated over a square gate of duration W (See description in ref. 2 in the Bibliography page). The user needs to provide the gate positions where the known minimum (Min Gate) and maximum (Max Gate) of the decay are observed (best obtained by looking them up on a reference decay with high SNR). The analysis computes the baseline value (Background), Amplitude and Lifetime associated with the decay and displays them on the top right of the Fluorescence Decay panel. These parameters as well as the total integrated intensity are associated with the decay and the corresponding phasor plot data point (Phasor Graph panel).

3.3. Background file method

Another method consist in choosing a time-gated data file (preferably acquired using the same settings as the data to be analyzed) corresponding to pure background signal. Selection of the File method will show a Background File selection box, with which this file can be selected. Subsequently, this file's data will be subtracted from new datasets, pixel-by-pixel and gate-by-gate, prior to processing. The Background Scaling Factor parameter (default value: 1) can be used to adjust the amount of background file correction. Using a value larger F, each gate image i will be subtracted the corresponded background gate image i times F. This allows adjusting for different acquisition parameters between data and background acquisition.

Important Note: When the background file subtraction method is selected, any change to the background file (or switching from a different method to the background file subtraction method) requires the current datafile to be reloaded. Indeed file background subtraction happens when the dataset is loaded.

Warning: Note that when the background file subtraction method is selected, the position of the Background subtraction method in the Operation Order list is ignored (it is always performed first).

4. Reverse Gates (*)

When selected it, changes the direction of the plotted decay, so that the tail of the decay comes after the rising part. This option is needed for some SwissSPAD datasets.

5. Normalization

Checking the Normalize Decay checkbox will apply a simple normalization (division by the maximum decay value) to each decay before display.

6. Decay Smoothing

Occasionally, a decay may be affected by undesirable “spikes”. It is sometimes possible to remove those spikes using cubic basic spline smoothing (details can be found at http://zone.ni.com/reference/en-XX/help/371361P-01/gmath/cubic_spline_fit/). The Cubic Spline Fit implementation of LabVIEW is used without weights, and smoothness parameters identically equal to 1 for all points, and balance parameter equal to $1 - 10^{-(x)}$, where x is the Smoothing Parameter defined in the Settings>>Fluorescence Decay>>Basic Analysis panel. From the Cubic Spline Fit description page linked to above:

If $x = 0$, the cubic spline fit is equivalent to a linear fit. If $x = \text{Inf}$, the cubic spline fit interpolates between the data points. If $x < 0$, an appropriate value is automatically calculated according to the time axis values.

To use this algorithm as part of the decay pre-processing, check the Smoothen Decay checkbox. The only exposed parameter for this algorithm, Smoothing Parameter, is accessible in the Settings>>Fluorescence Decay panel. Alternatively, an existing decay can be post-processed (creating a new curve) using the Smooth Plot function of the Context Menu (see below).

7. Decay Shift

Decays can occasionally “shift” along the time axis due to several possible causes (in general, setup instabilities). While this is normally not causing problems if data is properly calibrated, it is possible to force alignment of all decays along the time axis by checking the Shift Decay checkbox. There are several options associated with this functionality.

Type: this drop-down list gives access to 4 modes described below:

Rotate: this checkbox specifies whether the shift results in a rotation of the decay (considered periodic) or whether to pad the decay with zeros and discard points corresponding to negative abscissa. Shift: this parameter has different interpretation depending on the type of shift selected (see below for details) and is not always visible. Threshold: this parameter is used in the Threshold mode only (see below for details).

Decay shift modes details

Default: in this case, a constant shift is applied to all decays. This can for instance be useful to align the peak of a given sample to the zero point, or align decays acquired with different setups, etc. CFD: the constant fraction discrimination mode applies a constant shift to each decay before inverting it (multiplying it by -1) and adding it to the original decay. The effect of this operation, provided the shift is of the order of the IRF width or smaller, is to create a curve looking like a “chirp”, with a positive bump followed by a negative one, with a zero point in between. This point is generally stable if the shape of the decay is relatively constant (the amplitude can vary). The position of the zero-crossing point is then compared to that of a Reference Decay (see section 9 below for details on how to define a Reference Decay) and the difference between these two positions is defined as the decay shift. Threshold: in this mode, the provided Threshold parameter (the Shift parameter is not available) is used to find the first location in the decay where this threshold is crossed (from below). This location is compared to that obtained for the Reference Decay (see section 9 below for details on how to define a Reference Decay) and the difference between these two positions is defined as the decay shift. Cross-Correlation: in this mode, the CC of the decay and the Reference Decay (see section 9 below for details on how to define a Reference Decay) is computed and the position of its maximum determined and returned as the decay shift.

At the end of a series of decay analysis, it is possible to plot the calculated shifts in the Lifetime Graph of the Lifetime Analysis panel, using the Plot Decay Shifts context menu item in that graph.

8. Decay Straightening

Occasionally, samples can photobleach (or photobrighten) during the course of a series of gate acquisition. This phenomenon is identifiable by the fact that the recorded gate value at the end of a complete laser period is different (generally smaller but sometimes larger) than at the beginning of the period. The straighten function assumes that this is due to an exponential decay (or increase) of the signal due to some underlying uncontrollable phenomenon, and attempts to calculate the time scale of this variation as well as its amplitude, and finally, correct for it accordingly throughout the gate series. Up until version 0.18.1, this correction was applied as follows:

where $\{F_{\min}\}$ is the minimum decay value, T is the period, τ is the photobleaching/photobrightening time constant obtained from:

The sign of τ obtained from the above equation handles both cases. After version 0.18.2, the correction does not consider that the decay consists of a constant background ($\{F_{\min}\}$) added to a photobleaching/photobrightening component, as this background component should be taken care of by the background subtraction step, which usually precedes (*) all other pre-processing steps (except pile-up correction). As a consequence, the formula becomes:

where τ is given by: This equation requires that $F(0)$ and $F(T)$ be non-zero. (*) Note that since version 0.19, it

is possible to change the order of the different decay pre-processing operations (except pile-up correction, which remains the first operation). This means that if background correction follows decay straightening, the assumption of the straightening algorithm may be incorrect (i.e. the algorithm will assume that both decay and background exponentially increase or decrease with the same time constant).

9. Decay Extrapolation

In case the decay tail doesn't reach the background level, the resulting phasor will be offset by an amount that will depend on the final value reached by the decay. It is possible to compensate artificially for this truncation by extrapolating the decay with an exponential tail. The parameters defining the range of this extrapolation are defined in Settings>>Fluorescence Decay>>Basic Analysis: The Tail Fraction parameter specifies what fraction of the decay (starting from the end) is used to perform a fit to a single exponential decay. The Additional Points parameter specifies by how many points (spaced as in the original decay) to add to the decay. The Extrapolate Decay checkbox only applies when computing decays from scratch. If checked, the computed decays are automatically extrapolated.

10. Other Decay Processing Functions

Other decay processing functions are accessible via the Decay Graph context menu. Most of the context menu items are self-explanatory. Items are grouped in different categories of functions:

Edit plot Visibility (and plot style) functions Deletion functions Save functions Load Plot(s) Plot processing
Fitting functions IRF-related functions Graph copy/export/visibility functions

As for all graphs in AlliGator, the checkboxes in front of plot names in the graph legend have a dual function. When checked, the plot is visible AND selected. When unchecked, the plot is hidden AND deselected.

Single plot functions can be used by right-clicking on the plot of interest in the graph or its legend. Note that the Export menu is a bit different in this respect: to export a single plot to the clipboard as an ASCII formatted data set, right-click on that plot's legend (the graphic part of it). To export the WHOLE graph (including hidden plots), right-click in the graph region. Selected plots (or individual plots) can be directly saved in an ASCII file using the Save functions of the above menu.

10.1. Edit Plot

This menu item opens the plot on which the user has right-clicked in the Plot Editor, where several basic operations can be performed.

Warning: the edited plot replaces the original plot unless the Copy button is pressed. It is possible to cancel the operation at any time while in the Plot Editor.

Plot Editor functionalities are described in the corresponding page of the manual.

10.2. Process Plot submenu

$y \gg f(y)$ Transform: selecting this item opens up a dialog window to enter an algebraic formula:

The corresponding amplitude values of the plot (y) will be modified and replaced by y' as defined by the formula (assuming that the syntax is correct. For a list of supported functions, please refer to this LabVIEW help page).

$(x, y) \gg (f, g)(x, y)$ Transform: selecting this item opens up a dialog window to enter an algebraic formula:

The corresponding time (x) and amplitude (y) values of the plot will be modified and replaced by (x', y') as defined by the formulas (assuming that the syntax is correct. For a list of supported functions, please refer to this LabVIEW help page).

Smoothen Plot: see the Decay Smoothing section above for a description of this operation. The difference is that this menu item allow performing this operation after the decay as been computed and displayed, rather than at the time it is extracted from the dataset. Denoise Plot: When smoothing by cubic spline provides unsatisfactory results, the Denoise Plot function may provide better results. This function relies on wavelet analysis routines of the LabVIEW Advanced Signal Processing Toolkit, detailed at http://zone.ni.com/reference/en-XX/help/371419D-01/lvwavelettk/wa_de_noise/. Parameters can be set in the Settings>>Fluorescence Decay>>Advanced Analysis tab:

Average Selected Plots: This function does what it says and creates an additional plot. Compute Cumulative Function: This function can be useful to define what fraction of a decay to fit, as discussed in doi: 10.1117/1.3469797. Straighten Decay: see the Decay Straightening section above for a description of this operation. The difference is that this menu item allow performing this operation after the decay as been computed and displayed, rather than at the time it is extracted from the dataset. Extrpolate Decay: see the Decay Extrapolation section above for a description of this operation. The difference is that this menu item allow performing this operation after the decay as been computed and displayed, rather than at the time it is extracted from the dataset.

1.6.5 Fluorescence Decay Fitting

AlliGator offers basic capabilities to analyze data using standard techniques of fluorescence decay fitting:

- Individual decays can be fitted with a single or double-exponential decay model convolved with an experimental instrument response function (IRF).
- Multiple ROIs in the image source can be fitted instead of their phasor calculated, the resulting parameters being output as maps (see *Decay Fit Parameter Map Panel*)
- Time series analysis of a ROI in the image source can use “Series Fit” instead of “Series Phasor” (each time point decay is fitted with the same model, and selected parameters are output as time-series).
- Selected decays can be globally fit to a single or double-exponential decay convolved with an experimental IRF, some parameters of the model being common to all decays.

These different capabilities are described next.

Single decay fit

Overview

A single plot can be fitted by a model function convolved with the selected IRF by right-clicking on its legend (or close to it in the graph) and selecting Fit Decay. The parameters specifying the type of fit and constraints used are defined in the Settings >> Fluorescence Decay >> Fit Options panel reproduced below:

<insert figure here>

Several fit options, discussed below, are available:

- Model
- Method
- Algorithm
- Weighted Fit
- Show Fit Residuals
- Type of Residuals
- Max & Min Decay Percentile
- Show Full Decay
- Parameter Uncertainties
- Periodic Boundaries
- Termination Criteria
- Use Local IRF

Parameter constraints, guess parameters and whether or not to display the output as plots are managed in the Settings >> Fluorescence Decay >> Fit Parameters panel

- **Fit with constraints applied on individual parameters is handled by an array of Fit Parameter Constraints specifying the:**
 - Parameter
 - Min & Max Value
 - whether or not it is a Global parameter (applicable in case of multiple fits only)
 - whether the constraint is used or not
- **Guess parameters can be provided in the corresponding array by selecting the parameter name and providing the guess value. Additionally, two options can be turned on or off:**
 - Use Guess Parameters
 - Use Last Fitted Parameters
- The Displayed Fit Parameters array only applies to time-series analysis and will be discussed in that context in a later section.

Fit Options

- **Fit Model:** Two models are currently available.

A single exponential (1-Exponential) model defined by:

$$f(t) = A_1 \exp\left(-\frac{t}{\tau_1}\right) + b$$

where b is the baseline and the IRF is offset by an amount (i.e. centered at) t_0 .

A double exponential (2-Exponentials) model defined by:

$$f(t) = A_1 \exp\left(-\frac{t}{\tau_1}\right) + A_2 \exp\left(-\frac{t}{\tau_2}\right) + b,$$

- **Fit Weight:** Two types of fits can be performed:

an unweighted fit where all data points are equally weighted in the minimization function (sum of difference squared), or a weighted fit, where each data point i is weighted by $1/f_i$ (or 1 if $f_i = 0$), where f_i is the function value.

- **Residuals:** The fit residuals (difference between the original decay and its fit) can be optionally plotted in addition to the fit itself. Several weighting schemes can be chosen:

<insert figure here>

The standard residual is the mere difference between the original decay and its fit, while the normalized residual is the difference divided by the function value. The reduced residual is the difference divided by the square root of the absolute value of the function value.

- **Min & Max Decay Percentile:** The fit can be performed over the whole decay or limited to the “tail” part of the decay. The latter is defined as the part of the decay located between XX% of the decay maximum (max percentile) and YY% ($0 \leq YY < XX \leq 1$) of the decay maximum (min percentile).
- **Show Full Decay:** When only part of the decay is fitted, it is possible to show the fitted curve (and residuals, optionally) calculated over the full decay range by checking this checkbox. The default (unchecked) is to only show the decay over the selected range.
- **Parameter Uncertainties:** Because parameter uncertainty calculation involves computing the covariance matrix of all parameters, this can be very memory consuming in the case of global fit of large data sets. In that case, it might be desirable to skip calculation of parameter uncertainties by leaving this checkbox unchecked.

- Periodic Boundaries: This option is available in beta mode and allows enforcing periodic boundary conditions (and provide a value for the repetition period).

<insert figure here>

This is mostly useful for large gates (e.g. SwissSPAD data) for which the resulting decay does not look anymore as a sharp rise followed by a tail decaying to background level, but instead as a continuous “wave”. In these conditions, it is advantageous to treat the decay as periodic. Note that the recorded decay needs to be no longer than the provided period for the fit to be any good (it can be shorter, i.e. truncated). This option has no effect on global fits at this time.

Fit Parameter Constraints

Fit parameters can be constrained within a specified range defined by the min (-Inf if unconstrained) and max value (Inf if unconstrained).

The list of actual parameters that can be constrained depends on the chosen model:

For instance, choosing τ_2 as a constrained parameter in a 1-Exponential model will have no effect.

If a parameter is unconstrained, it is possible to remove it from the array of constrained parameters by right-clicking on it and choosing Delete Element. If no parameter is constrained, it is possible to delete all elements of the array by right-clicking on the scrollbar and choosing Empty Array.

Fit Results

In addition to the plot output(s) in case of a successful fit, the fit results are output to the Notebook. A typical output will read

```
2-Exponentials unweighted fit of XXXXX
Fit range: 0%-100%
Fitted Parameters:
Offset: 0.003166 ± 0.001487 [-0.1, 0.1]
Baseline: -0.003543 ± 0.004714
A1: 0.26873 ± 0.140506 [0, Inf]
tau 1: 0.446588 ± 0.151041 [0.005, 1]
A2: 1.122693 ± 0.144699 [0, Inf]
tau 2: 1.13671 ± 0.078737 [0, 2]
R^2: 0.999408
Reduced Chi^2: 0.007177
Standard residuals
```

where XXXXX is the decay name. $\{R^2\}$ and the reduced $\{\chi^2\}$ as well as the 68% confidence intervals (errors) are defined according to the definitions provided here. If the fit fails, an error message will be displayed instead (and not plot added to the Decay Graph).

Time-series decay fit

In the case of a time-series analysis, decay fits can be performed instead of phasor analysis, by choosing Time-Series Analysis (Fitting) in the Analysis menu:

<insert figure here>

Each time point decay is fitted separately, following the protocol described previously for single decays. In addition, it is possible to generate one or more plots of the evolution of selected fit parameters across the series, using the Displayed Fit Parameters array. These plots will be output in the Lifetime Graph of the Lifetime Analysis panel (see corresponding manual page). Parameters that can be displayed can be chosen from the following list:

<insert figure here>

This list includes the fit parameters and derived quantities, such as the mean lifetime $\langle\tau\rangle$ or fraction f_1 and f_2 (for the 2-Exponentials model, defined below), or the R^2 and χ^2 outputs.

$$\langle\tau\rangle = \frac{A_1\tau_1 + A_2\tau_2}{A_1 + A_2}$$

$$f_1 = \frac{A_1\tau_1}{A_1\tau_1 + A_2\tau_2}$$

$$f_2 = 1 - f_1$$

1.7 Alligator Reference

The following pages describe specific topics not covered in other parts of the manual.

1.7.1 AlliGator HDF5 File Format

It is possible to save data loaded in AlliGator (and reload them!) in a file format based on the HDF5 hierarchical format developed and maintained by the HDF Group. This page briefly describes the file format used by AlliGator. Note that the latest AlliGator version supports all previous HDF5 file format versions.

v 0.7 (2023-09-02)

This version adds new information to the *SwissSPAD Detector Information* group, and a new parameter to the *DAQ Parameters* cluster (and renames some of the parameters).

- The SwissSPAD-detector specific group, *SwissSPAD Detector Information*, is comprised of 8 strings and three Boolean flags (stored as integers) as shown below:

Parameter Name	Type	Parameter Description
Sensor Type	String	2 types are currently supported: SS2 and SS3
Microlens	Boolean	Specifies whether the detector is equipped with microlenses
Detector PCB Version	String	4 types are currently supported, namely <code>unknown</code> , <code>SS2_PCB1</code> , <code>SS2_PCB2</code> , <code>SS3_PCB1</code>
Bottom Half	Boolean	Specifies whether the bottom half was used
Bottom FPGA Serial Number	String	as obtained from the Opal Kelly API
Bottom Bitfile Path	String	Path of the bitfile used during acquisition
Bottom Bitstream Version	String	as obtained from the bitfile header
Top Half	Boolean	Specifies whether the bottom half was used
Top FPGA Serial Number	String	as obtained from the Opal Kelly API
Top Bitfile Path	String	Path of the bitfile used during acquisition
Top Bitstream Version	String	as obtained from the bitfile header

- The *DAQ Parameters* cluster (saved as a Table):

Parameter Name	Unit	Parameter Description
# Pixel X		number of pixels in the horizontal direction
# Pixel Y		number of pixels in the vertical direction
# Gates		number of time points in the recorded decays
# Dataset:		for time-series, number of time points. The default is a single dataset
Gate Image Exposure	s	Time during which photon are accumulated. For time-gated acquisition, this corresponds to a fraction (duty cycle) of the actual camera integration per gate. For a TCSPC dataset, this correspond to the pixel dwell time
Macro-time Gate Separation	s	time separation between gate acquisition in “laboratory” time (macrotime) For instance, in a camera-based acquisition, this time could be equal to the Exposure/Gate (s) parameter (if readout is performed while the next gate is being acquired), or longer (if readout adds a delay until the next gate image is acquired. For TCSPC data, this value will typically be 0. When this information is unavailable, it takes the value NaN (not a number)
Nanotime Gate Separation	s	Separation between the starting point of consecutive gates. For TCSPC data, this is equal to the Gate Duration (s) parameter described below. For a camera-based acquisition, this is the user-specified gate shift
Gate Width	s	Time during which photons are detected after each laser pulse. For TCSPC data, this quantity is identical to the Nanotime Gate Separation (s) parameter above and corresponds to the TCSPC bin size. For a camera-based time-gated acquisition, this will usually be a much larger value (typically several 100 ps to several ns)
Laser Period	s	Laser Period used during acquisition
SYNC Period	s	Sync pulse period used to trigger the recording. This will typically be equal or larger than (multiple of) the laser period, such as for a TAC/TDC window duration
Gate Image Integration	s	integration time needed to acquire a single gate image. For time-gated acquisition, this is generally larger than the gate exposure. For a TCSPC dataset, this correspond to the pixel dwell time

v 0.6.1 (2022-07-27)

This minor version adds a new field (*Dataset Timestamp*, a double-precision floating point number) to the *File Information* group.

This parameter (unit: s) records the start of the recording of the dataset in the file (in a laboratory clock).

v 0.6 (2022-04-13)

This version supports multiple gate images per gate step (as is necessary for the SwissSPAD3 detector).

One new field (*Gate Names*, a string array) has been added to the *File Information* group, resulting in the following structure:

- *File Information* group containing 9 strings, 1 string array, 2 integers and 1 Boolean:

Parameter Name	Type	Parameter Description
File Type	String	currently, a single file type is recognized, named Wide-Field Time-Gated Data
File Version	String	for a file compliant with this version, the string would be 0.5
Gate Names	String Array	Root names of the different types of gates. For single gate image per gate step (the default case), this array will contain the single string <i>Gate</i>
Compression	Boolean	TRUE if the data was compressed, FALSE it was not
Author	String	user provided information
MAC Address	String	unique computer identified
Windows Username	String	in case the Author field was not filled
Creation Date & Time	String	typically of the form Date: Month/Day/Year, Time: Hours:Minutes:Seconds AM/PM. The format is not strict, the field being provided for info only
Data Type	String	U8. U16 or SGL depending on whether individual gate images are 8-bit, 16-bit or single-precision floating point image. This field should be read before interpreting each image's data
File Path	String	original file path
Sample Information	String	user provided information
# Datasets in Series	I32 ³	if the file is part of a dataset series, this number is different from 1 and specifies the number of datasets in the series
Dataset ID in Series	I32	if the file is part of a dataset series, this number can be different from 1 and specifies the order of the dataset in the series

It also adds a SwissSPAD-detector specific group, *SwissSPAD Detector Information*, comprised of 5 strings as shown below:

- *SwissSPAD Detector Information* group containing 5 strings:

¹ Booleans are stored as integers in HDF5 files (FALSE: 0, TRUE: 1)

³ I*x*: x-bit signed integer (e.g. I16: 16-bit signed integer)

Parameter Name	Type	Parameter Description
Sensor Type	String	2 types are currently supported: SS2 and SS3
Detector PCB Version	String	4 types are currently supported, namely unknown, SS2 PCB1, SS2 PCB2, SS3 PCB1
FPGA Serial Number	String	as obtained from the Opal Kelly API
Bitfile Path	String	Path of the bitfile used during acquisition
Bitstream Version	String	as obtained from the bitfile header

- *DAQ Parameters* cluster is unchanged.
- *Metadata* string is unchanged
- *Gate Images* group now contains possibly more than one gate image per gate step. Each is a 2D arrays labeled *Gate_Name* *n*, where *n* will typically run from 1 to *#Gates* (the parameter found in the *DAQ Parameters* cluster described above) and *Gate_Name* is one of the strings found in the *Gate Names* string array found in the *File Information* group. For instance, if *Gate Names* = [Bottom INT Gate, Bottom G2 Gate], the gate image names will be of the type: Bottom INT Gate 1, Bottom G2 Gate 1, Bottom INT Gate 2, Bottom G2 Gate 2, etc.

v 0.5 (2021-04-27)

Two new fields (*MAC Address* and *Windows Username*) have been added to the *File Information* group, while an additional *SYNC Period* parameter was added to the *DAQ Parameters* cluster. The resulting structure is thus:

- *File Information* group containing 9 strings, 2 integers and 1 Boolean:

Parameter Name	Type	Parameter Description
File Type	String	currently, a single file type is recognized, named Wide-Field Time-Gated Data
File Version	String	for a file compliant with this version, the string would be 0.5
Compression	Boolean ^P	TRUE if the data was compressed, FALSE it was not
Author	String	user provided information
MAC Address	String	unique computer identified
Windows Username	String	in case the Author field was not filled
Creation Date & Time	String	typically of the form Date: Month/Day/Year, Time: Hours:Minutes:Seconds AM/PM. The format is not strict, the field being provided for info only
Data Type	String	U8. U16 or SGL depending on whether individual gate images are 8-bit, 16-bit or single-precision floating point image. This field should be read before interpreting each image's data
File Path	String	original file path
Sample Information	String	user provided information
# Datasets in Series	I32 ^{Page 45,}	if the file is part of a dataset series, this number is different from 1 and specifies the number of datasets in the series
Dataset ID in Series	I32	if the file is part of a dataset series, this number can be different from 1 and specifies the order of the dataset in the series

- *DAQ Parameters* cluster (saved as a Table):

Parameter Name	Unit	Parameter Description
# Pixel X		number of pixels in the horizontal direction
# Pixel Y		number of pixels in the vertical direction
# Gates		number of time points in the recorded decays
# Dataset:		for time-series, number of time points. The default is a single dataset
Exposure/Gate	s	integration time needed to acquire a single gate image. For time-gated acquisition, this corresponds to the actual camera integration per gate. For a TCSPC dataset, this correspond to the pixel dwell time
Macro-time Gate Separation	s	time separation between gate acquisition in “laboratory” time (macrotime) For instance, in a camera-based acquisition, this time could be equal to the Exposure/Gate (s) parameter (if readout is performed while the next gate is being acquired), or longer (if readout adds a delay until the next gate image is acquired. For TCSPC data, this value will typically be 0. When this information is unavailable, it takes the value NaN (not a number)
Nanotime Gate Separation	s	Separation between the starting point of consecutive gates. For TCSPC data, this is equal to the Gate Duration (s) parameter described below. For a camera-based acquisition, this is the user-specified gate shift
Gate Duration	s	Time during which photons are detected after each laser pulse. For TCSPC data, this quantity is identical to the Nanotime Gate Separation (s) parameter above and corresponds to the TCSPC bin size. For a camera-based time-gated acquisition, this will usually be a much larger value (typically several 100 ps to several ns)
Laser Period	s	Laser Period used during acquisition
SYNC Period	s	Sync pulse period used to trigger the recording. This will typically be equal or larger than (multiple of) the laser period, such as for a TAC/TDC window duration

- *Image Information* group contains 2 clusters:

Parameter Name	Type	Parameter Description
Image ROI Information	Cluster:	
Save ROI Only	Boolean	whether a ROI (TRUE) or the full sensor date (FALSE) is saved
Left	U16 ²	left edge coordinate of the rectangular ROI in the full sensor reference frame
Top	U16	top edge coordinate of the rectangular ROI in the full sensor reference frame
Right	U16	right edge coordinate of the rectangular ROI in the full sensor reference frame
Bottom	U16	bottom edge coordinate of the rectangular ROI in the full sensor reference frame
Use Current ROI	Boolean	whether the ROI corresponds to a user drawn ROI (TRUE) or was entered manually (FALSE)
Image Binning Options	Cluster	
Use Image Binning	Boolean	whether binning is used (TRUE) or not (FALSE)
X Bin	I32	binning factor in the horizontal direction
Y Bin	I32	binning factor in the vertical direction

- *Gate Images* group containing the gate images in the dataset as a series of 2D arrays labeled *Gate n*, where *n* will typically run from 1 to *# Gates* (the parameter found in the *DAQ Parameters* cluster described above).
- *Metadata* string contains any other information that does not fit in any of the previous fields. There is currently no specified formatting.

v 0.4 (2021-02-27)

A new field (*Compression*) was added to the *File Information* group to specify whether compression was used (skipping compression speeds up data saving and loading). In principle, this should have no impact on HDF5 loading, as the HDF5 library is capable of determining this without user input. This field was added for information only.

- *File Information* group containing 7 strings, 2 integers and 1 Boolean:

² U*x*: x-bit unsigned integer (e.g. U16: 16-bit unsigned integer)

Parameter Name	Type	Parameter Description
File Type	String	currently, a single file type is recognized, named Wide-Field Time-Gated Data
File Version	String	for a file compliant with this version, the string would be 0.4
Compression	Boolean ^P	TRUE if the data was compressed, FALSE it was not
Author	String	user provided information
Creation Date & Time	String	typically of the form Date: Month/Day/Year, Time: Hours:Minutes:Seconds AM/PM the format is not strict, the field being provided for info only
Data Type	String	U8. U16 or SGL depending on whether individual gate images are 8-bit, 16-bit or single-precision floating point image. This field should be read before interpreting each image's data
File Path	String	original file path
Sample Information	String	user provided information
# Datasets in Series	I32 ^{Page 45,}	if the file is part of a dataset series, this number is different from 1 and specifies the number of datasets in the series
Dataset ID in Series	I32	if the file is part of a dataset series, this number can be different from 1 and specifies the order of the dataset in the series

Additional information on the part of the sensor data saved in the file, a new *Image Information* group has been added as described below.

- *Image Information* group contains 2 clusters:

Parameter Name	Type	Parameter Description
Image ROI Information	Cluster:	
Save ROI Only	Boolean	whether a ROI (TRUE) or the full sensor date (FALSE) is saved
Left	U16 ^{Page 49, 2}	left edge coordinate of the rectangular ROI in the full sensor reference frame
Top	U16	top edge coordinate of the rectangular ROI in the full sensor reference frame
Right	U16	right edge coordinate of the rectangular ROI in the full sensor reference frame
Bottom	U16	bottom edge coordinate of the rectangular ROI in the full sensor reference frame
Use Current ROI	Boolean	whether the ROI corresponds to a user drawn ROI (TRUE) or was entered manually (FALSE)
Image Binning Options	Cluster	
Use Image Binning	Boolean	whether binning is used (TRUE) or not (FALSE)
X Bin	I32	binning factor in the horizontal direction
Y Bin	I32	binning factor in the vertical direction

- *Gate Images* group containing the gate images in the dataset as a series of 2D arrays labeled Gate *n*, where *n* will typically run from 1 to # *Gates* (the parameter found in the *DAQ Parameters* cluster described above).
- *Metadata* string contains any other information that does not fit in any of the previous fields. There is currently no specified formatting.

v 0.3 (2021-01-29)

In order to support asynchronous data saving by third party software, version 0.3 changes the way the individual gate images are saved. Instead of being all clumped into a monolithic 3D array, each gate image is saved as a separate 2D array, as described below. This has the additional advantage that it simplifies accessing individual gates. As in v 0.2, each image gate is compressed, which significantly saves space, without incurring any significant decoding time cost.

The file is comprised of:

- *File Information* group containing 7 strings and 2 integers:

Parameter Name	Type	Parameter Description
File Type	String	currently, a single file type is recognized, named Wide-Field Time-Gated Data
File Version	String	for a file compliant with this version, the string would be 0.4
Author	String	user provided information
Creation Date & Time	String	typically of the form Date: Month/Day/Year, Time: Hours:Minutes:Seconds AM/PM the format is not strict, the field being provided for info only
Data Type	String	U8. U16 or SGL depending on whether individual gate images are 8-bit, 16-bit or single-precision floating point image. This field should be read before interpreting each image's data
File Path	String	original file path
Sample Information	String	user provided information
# Datasets in Series	I32 ^{Page}	if the file is part of a dataset series, this number is different from 1 and specifies the number of datasets in the series
Dataset ID in Series	I32	if the file is part of a dataset series, this number can be different from 1 and specifies the order of the dataset in the series

- *DAQ Parameters* cluster (saved as a Table): this is unchanged since v 0.1 and is reproduced here for convenience:

Parameter Name	Unit	Parameter Description
# Pixel X		number of pixels in the horizontal direction
# Pixel Y		number of pixels in the vertical direction
# Gates		number of time points in the recorded decays
# Dataset		for time-series, number of time points. The default is a single dataset
Exposure/Gate	s	integration time needed to acquire a single gate image. For time-gated acquisition, this corresponds to the actual camera integration per gate. For a TCSPC dataset, this correspond to the pixel dwell time.
Macro-time Gate Separation	s	time separation between gate acquisition in “laboratory” time (macrotime). For instance, in a camera-based acquisition, this time could be equal to the Exposure/Gate (s) parameter (if readout is performed while the next gate is being acquired), or longer (if readout adds a delay until the next gate image is acquired. For TCSPC data, this value will typically be 0. When this information is unavailable, it takes the value NaN (not a number).
Nanotime Gate Separation	s	Separation between the starting point of consecutive gates. For TCSPC data, this is equal to the Gate Duration (s) parameter described below. For a camera-based acquisition, this is the user-specified gate shift.
Gate Duration (s)	s	Time during which photons are detected after each laser pulse. For TCSPC data, this quantity is identical to the Nanotime Gate Separation (s) parameter above and corresponds to the TCSPC bin size. For a camera-based time-gated acquisition, this will usually be a much larger value (typically several 100 ps to several ns).
Laser Period (s)	s	Laser Period used during acquisition

- *Gate Images* group containing the gate images in the dataset as a series of 2D arrays labeled `Gate n`, where n will typically run from 1 to *# Gates* (the parameter found in the *DAQ Parameters* cluster described above).

Note: there is no requirement that the number of gate images stored in the file matches the number of gates provided by the *# Gates* parameter. For instance, if an acquisition is interrupted before the total number of gates has been acquired, the file will be closed without the full complement of gates. There is also no check that the number of gate images remains smaller or equal to that parameter. User beware!

- *Metadata* string contains any other information that does not fit in any of the previous fields. There is currently no specified formatting.

v 0.2

HDF5 files saved in this version are identical to v 0.1 files except that the *Gate Images* data is compressed using the *deflate* algorithm, with a level 9 compression and block size equal to $1 \times 1 \times G$ where G is the number of gates.

v 0.1

The file is comprised of:

- *File Information* group containing two strings:

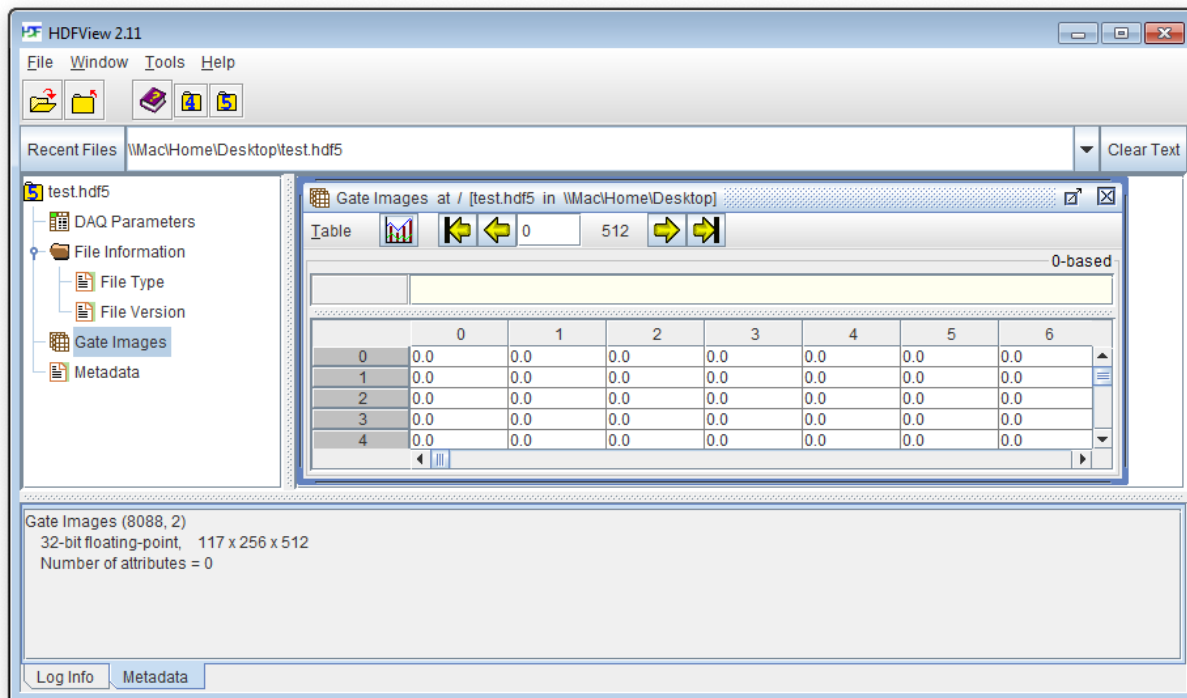
Parameter Name	Parameter Description
File Type	currently, a single file type is recognized, named Wide-Field Time-Gated Data
File Version	for a file compliant with this version, the string would be 0.1

- *DAQ Parameters cluster* (saved as a Table)

Parameter Name	Unit	Parameter Description
# Pixel X		number of pixels in the horizontal direction
# Pixel Y		number of pixels in the vertical direction
# Gates		number of time points in the recorded decays
# Dataset		for time-series, number of time points. The default is a single dataset
Exposure/Gate	s	integration time needed to acquire a single gate image. For time-gated acquisition, this corresponds to the actual camera integration per gate. For a TCSPC dataset, this correspond to the pixel dwell time.
Macro-time Gate Separation	s	time separation between gate acquisition in “laboratory” time (macrotime). For instance, in a camera-based acquisition, this time could be equal to the Exposure/Gate (s) parameter (if readout is performed while the next gate is being acquired), or longer (if readout adds a delay until the next gate image is acquired. For TCSPC data, this value will typically be 0. When this information is unavailable, it takes the value NaN (not a number).
Nanotime Gate Separation	s	Separation between the starting point of consecutive gates. For TCSPC data, this is equal to the Gate Duration (s) parameter described below. For a camera-based acquisition, this is the user-specified gate shift.
Gate Duration (s)	s	Time during which photons are detected after each laser pulse. For TCSPC data, this quantity is identical to the Nanotime Gate Separation (s) parameter above and corresponds to the TCSPC bin size. For a camera- based time-gated acquisition, this will usually be a much larger value (typically seveal 100 ps to several ns).
Laser Period (s)	s	Laser Period used during acquisition

- *Metadata* string contains any other information that does not fit in any of the previous fields. There is currently no specified formatting.
- *Gate Images*: 3-dimensional array of single precision floating point numbers

A snapshot of a file as viewed with HDFView, the free HDF5 file viewer developed by the HDF Group is shown below:



Notes

1.7.2 Python Plugins

AlliGator supports user-provided Python functions which appear as right-click menu items for the corresponding object or menu.

Currently, only two objects support plugins: the *Decay Graph* and the *Source Image*. In addition, *FLI Dataset* plugins are supported as sub-menus of the **Analysis:FLI Dataset** menu. Since objects have associated menus (sometimes empty) in the **Analysis** menu, plugins can also be inserted in the **Analysis:Decay Graph** and **Analysis:Source Image** menus.

This page does **not** describe how to develop Python plugins for AlliGator, but provides an overview of the design principle, followed by a description of the mandatory features for a plugin. The last section show an example of plugin for the *Decay Graph* graph. Example of plugins for the different supported objects can be found in the example scripts provided by default with AlliGator.

Details on how to implement an AlliGator Python plugin can be found in the [AlliGator Python Plugins API](#) page.

Design Principle

LabVIEW has some basic built-in support for communication with Python in the form of the *Python Node*. The *Python Node* requires spawning an instance of Python (the path and version of which can be specified by the user in the **Settings:Plugins** window), which is done when AlliGator starts, and passing it a script path and function name within that script, together with parameters. The functions then returns a result or possibly an error code and message, in which case this is displayed in the **Notebook** window.

The type of data that is passed to a plugin function is in part specified in the function's body using AlliGator plugin 'statements' (described in subsequent sections), but some of the data also depends on the target of that plugin and does not need to be specified within the function (in other words, it is handled automatically).

For instance, a plugin for the *Decay Graph* will always pass a list of plots to the function (the so-called `graph_data_in`), the content of which depends on whether the plugin acts on all plots, selected plots or only the plot from which the function was invoked.

The exact nature of the default data passed to a function is described in the [AlliGator Python Plugins API page](#).

The non-default data passed to the function is specified by the developer in the *Input Parameters Definition* section, and is passed to the Plugin function as a dictionary (see below and [AlliGator Python Plugins API page](#) for details). Some of this data is provided by the user in a dialog window, which is automatically opened by AlliGator before executing the plugin function. The rest is specifically requested internal AlliGator data and does not require user input.

The data returned by the function is, as the input data, separated into two parts: the default data type for the function's target, and the plugin-specific data, specified by the user in an *AlliGator Output Value Type & Destination* section. As the input parameters specified in the *Input Parameters Definition* section, it is passed as a dictionary (see below and [AlliGator Python Plugins API page](#) for details).

As for the default input data, the default output data depends on the function's target and is described in the [AlliGator Python Plugins API page](#).

Both input and output definition sections are discussed below.

Plugins Folder

Plugins consists of functions saved within Python script files (.py extension).

These script files need to be saved within the Python Plugins folder located in the AlliGator installation folder (generally in `C:\Users\User_Name\AppData\Local\AlliGator`, unless specified otherwise during installation).

Each file can be saved within a subfolder hierarchy within the Plugins Folder.

Plugin Files

Each .py plugin file will give its name to a submenu in the right-click menu of the object it is targeting (see below to learn how to specify a plugin's target) or the **Analysis** submenu it is added to (same remark). For instance, if a script file name `UCLA_Plot_Functions.py` is added and targeted to the **Decay Graph**, the functions within will be added to the *Decay Graph* context menu in a submenu named `UCLA Plot Functions`.

For instance, if there are two files in Plugins folder named `plugins 1.py` and `plugins 2.py`, both targeted to the **Decay Graph**, two submenus will be added to the right-click menu of the **Decay Graph**, respectively called `plugins 1` and `plugins 2`.

If a submenu already exists, the plugin functions within the .py file will be added to it.

If a function with the same name already exists within the submenu, it will be added with the same name, but will be recognized as distinct by AlliGator. Because that could be confusing for a user, it is recommended to name .py files

(and thus plugin submenus) in a way that is fairly unique, for instance by including a lab name in it (as in the example above).

Each plugin submenu is separated from the previous one by an horizontal divider line.

Each plugin file can contain an arbitrary number of Python functions, with valid names for Python, say `function_1`, `function_2`, ..., `function_n`. The resulting submenu will contain as many items as there are functions, their names being the names of the functions. For instance, if `plugins_1.py` contains `n` functions: `function_1`, `function_2`, ..., `function_n`, the corresponding submenu will look like:

- plugins 1
 - function 1
 - function 2
 - ...
 - function n

The next section will describe naming conventions for plugin functions, and how to distinguish them from helper functions.

Plugin Function Names

Plugin functions appear in their corresponding submenus as they are named in the script file, with the following transformations:

- a single underscore at the beginning of the function name is interpreted as meaning that an horizontal separator will precede that function in the menu.
- double underscores are alternatively replaced by a left and right parenthesis , respectively preceded and followed by a space.
- single underscores are replaced by single spaces.

For instance, a function named `this_is_a_simple_function(*args, **kparams)` will appear as `this is a simple function` in the submenu.

Likewise, a function named `simple_operation__selected_plots__(*args, **kparams)` will appear as `simple operation (selected plots)` in the submenu.

Plugin Functions Special Syntax

Plugin functions need to contain some simple additional syntax to be usable by AlliGator. The additional elements appear within comments bracketed between triple hash tags (`###`) and should therefore not interfere with any Python-specific syntax.

These syntactic elements are as follows:

- `import` statements
- Destination
- Plugin Flag
- Input Parameters Definition
- Output Value Type & Destination

A plugin function may require special modules to perform its operations. These are typically imported at the beginning of the script. In addition to any plugin-specific module import statement, any AlliGator Python Plugin script requires the following additional two import statements:

```
import json
import alligator
```

The first is used to convert the input parameters string from JSON to a dictionary, and the output parameters dictionary to a JSON string, while the second refers to the `alligator.py` script provided in the Python Plugins folder, and contains type definitions that are useful to format input and output data for a plugin.

In addition to these two import statements, the other 4 elements to include are briefly reviewed next.

The first element (*Destination*) tells AlliGator where the plugin functions need to be inserted (in which menu or object's context menu). This is common to *all* functions in the script, therefore if plugins for different targets are developed (e.g. one function for the **Decay Graph**, and another for the **Source Image**), they will need to be in different script files.

The second element (*Plugin Flag*) specifies whether the function is actually a plugin function or an helper function (in other words, helper functions are unmodified Python functions and do not need any of the modifications discussed here).

The third element (*Input Parameters Definition*) is used to inform AlliGator about the parameters needed by the function (some parameters are passed by default, depending on the function's target, as discussed below). If no parameter is needed, this section can be ignored.

The last element (*Output Value Type & Destination*) is used by AlliGator to decide where to send the function's output. If no output is returned, this section can be ignored.

The syntax of these different elements is discussed in the following sections.

Note: In addition to these 4 mandatory elements, it is recommended to include a Python doc string to provide a description of what the function is doing, as well as information on whether or not and what user-provided parameters may be required. This doc string can be sent to the **Notebook** window by holding the H key down while selecting the Python plugin function in the corresponding menu (the function will not be executed).

Destination

To instruct AlliGator to insert a script's functions into a specific menu, the following statement needs to be inserted before the different functions:

```
### AlliGatorTarget = AlliGator/Object/Object_Name ###
```

and/or

```
### AlliGatorTarget = AlliGator/Menu/Object_Name ###
```

where `Object_Name` is the name of the target (e.g. `Decay Graph`, `Source Image` or `FLI Dataset`).

For instance, to insert all plugin functions within a script into the `Analysis:Decay Graph` menu, the following statement will be needed:

```
### AlliGatorTarget = AlliGator/Menu/Decay_Graph ###
```

Plugin Flag

To tell AlliGator that the function is a Plugin function, the following statement needs to be included after the doc string (if a doc string is provided) and before the *Input Parameters Definition* statement:

```
### IsAlliGatorPythonPlugin ###
```

If a function does not contain this statement, it will not be included in any AlliGator menu and will therefore be invisible to an AlliGator user. This will typically be the case of helper functions called by the Plugin functions.

Input Parameters Definition

Some plugins do not require any input parameters. For instance, if a *Decay Graph* plugin computes the mean of a plot, the only required input is the plot itself, which would be passed automatically as default data for a Graph plugin (see *Python Plugins API*). In that case, the remainder of this section can be ignored.

Other plugins will require either user input parameters (e.g. a scaling factor) or AlliGator parameters (e.g. the phasor frequency). Both types of parameters are declared at the beginning of the function in a custom comment section with the general structure shown below.

```
### AlliGator Input Parameters Definitions ###
### parameter_1:type_1 # description 1
...
### parameter_n:type_n # description n
### End of AlliGator Input Parameters Definitions ###
```

The first and last line of this (triple) commented section are used by AlliGator to find it and should be reproduced as shown above.

Single (or double) commented lines (or empty lines) in between these two lines will be ignored.

Input parameter declarations follow the model indicated above, that is, a triple comment symbol (###) followed a single parameter declaration per line.

A parameter declaration consists of a unique name (at least in a given function) , followed by a semicolon, followed by the parameter type (from the limited list of supported types (see *Python Plugins API* for details), including `AlliGator`, which indicates an internal AlliGator parameter. For a user-provided input parameter, a short *description* is recommended and should be provided as a single commented string (`# description n` being replaced by something more... descriptive!). It will appear next to the name and value of the parameter in a dialog box when the plugin is called. An example of such a dialog box (corresponding to the example shown at the bottom of this page) is provided below.

To expose which internal AlliGator parameters can be passed as input parameters, use the *Send* button in the **Settings:Plugins** panel. This will copy a list of these parameters in the clipboard, which can then be pasted in any text editor.

Check the *Parameter Names only* checkbox in the **Settings:Plugins** panel to get a JSON-formatted list of exposed internal AlliGator parameters without their current value.

As mentioned before, if no input parameter is needed by the function, this section can be omitted.

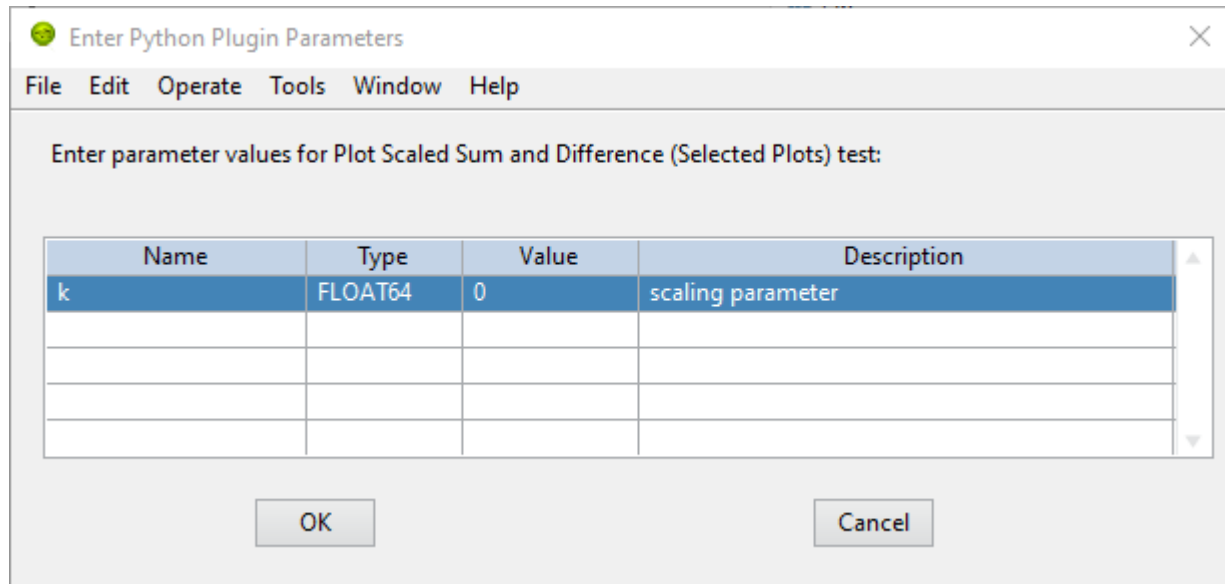


Fig. 1: AlliGator Python Pugin Input Parameters Dialog

Output Value Type & Destination

Python plugins outputs are of four general kinds (for details, see *Python Plugins API*):

- error code
- output message
- output sent to internal AlliGator variables
- output sent to objects

Error codes are internal to LabVIEW and will be formatted and sent to the **Notebook** if the **Verbose Error Mode** setting is on (**Settings:Miscellaneous**). Errors are generated by the Python script and are in general useful to debug a plugin. For this reason, it is recommended to turn on **Verbose Error Mode** when developing and testing a plugin.

The *output message* is an optional string generated by the plugin function. It is sent to the **Notebook**, and can therefore be used to display formatted results or information about the plugin's action (including a non-Python error message).

Output sent to internal AlliGator variables are a way to modify the internal state of AlliGator as a result of executing a Plugin. For instance, a plugin could be change a **Settings** flag, or the *phasor frequency*. Use with moderation, as this could have unintended consequences if done improperly.

All previous three types of output do not require any special syntax to be added to the plugin.

The only type of output parameter that requires inclusion of an *Output Value Type & Destination* comment section are values destined to one of AlliGator's displayed objects (Graph and Image). The syntax of this section, which should follow the *Input Parameters Definition* comment section is as follows:

```
### AlliGator Output Value Type & Destination ###
### value_type:destination
### End of AlliGator Output Value Type & Destination ###
```

The first and last line of this (triple) commented section are used by AlliGator to find it and should be reproduced as shown above.

Single (or double) commented lines (or empty lines) in between these two lines will be ignored.

Output value and destination statements follow the model indicated above, that is, a triple comment symbol (###) followed a single declaration per line.

Valid *value_type* are specified in the *Python Plugins API* page.

Valid *destination* statements depend on the *value_type* and the plugin's declared destination (so for instance, a *plots* value type cannot be sent to an image destination such as *Mask Image* but could be sent to *Decay Graph*).

If no output needs to be sent to a displayed object, this section can be omitted.

To better understand these different syntactic elements and how they fit into a Python function used as a plugin, it is easiest to look at the (not very useful as an AlliGator plugin) example installed with AlliGator and appearing as submenu item of *Analysis:Decay Graph* and as a context menu item of the *Decay Graph*.

Decay Graph example

The following example illustrates the principles outlined above. The same example with additional comments is installed in the *Plugins Folder* by default.

```
# Decay_Graph_Plugin_Example.py
# Example AlliGator Decay Graph Python Plugin

### AlliGatorTarget = AlliGator/Object/Decay Graph ###
### AlliGatorTarget = AlliGator/Menu/Decay Graph ###

import json
import alligator

def Plot_Scaled_Sum_and_Difference__Selected_Plots__test(
    graph_data_in, params_in_json, addtl_params_out_json_list):

    """Scaled Sum & Difference:

    Acts on the first two selected plots
    Expects one float parameter (k: float64)
    The resulting k*Sum and k*Difference plots are added to the Decay Graph
    """

    ### IsAlliGatorPythonPlugin ###

    ### AlliGator Input Parameters Definitions ###
    ### k:float64 # scaling parameter

    ### Phasor Frequency:AlliGator
    ### Reference Decay:AlliGator
    ### End of AlliGator Input Parameters Definitions ###

    ### AlliGator Output Value Type & Destination ###
    ### Plots:Decay Graph
    ### End of AlliGator Output Value Type & Destination ###

    message = 'Scaled Sum of Selected Plots and Phasor Frequency Update'
    exception_type = "None"
    exception_message = ""
```

(continues on next page)

(continued from previous page)

```

# decode the parameter string

params = json.loads(params_in_json)
k = params['k']
f = params['Phasor Frequency']

# decode the graph data named tuple

graph_name = graph_data_in.Graph_Name
plots = graph_data_in.Plots
nplots = len(plots)

# Adds and subtracts the first 2 plots if they have the same length
# otherwise returns an error

if nplots < 2:
    exception_type = "Error"
    exception_message = "Not enough selected plots!"
    graph_data_out = alligator.graph_plugin_data(
        Graph_Name = graph_name,
        Plots = [],
        Reference_Decay = alligator.empty_plot)
else:
    plot_data1 = plots[0]
    name1 = plot_data1.Plot_Name
    x1 = plot_data1.X_Array
    y1 = plot_data1.Y_Array
    plot_data2 = plots[1]
    name2 = plot_data2.Plot_Name
    x2 = plot_data2.X_Array
    y2 = plot_data2.Y_Array

    ref_decay_data = graph_data_in.Reference_Decay
    ref_decay_name = ref_decay_data.Plot_Name
    ref_decay_x = ref_decay_data.X_Array
    ref_decay_y = ref_decay_data.Y_Array

    # processing of the incoming data

    if (len(x1) != len(x2)):
        exception_type = "Error"
        exception_message = "Plots do not have the same length!"
        plots_out = []
    else:
        sumy = []
        for i in range(len(y1)):
            sumy.append(y2[i] + y1[i])
        diffy = []
        for i in range(len(y1)):
            diffy.append(y2[i] - y1[i])

    # we need to repack those plots into a list of named tuples

```

(continues on next page)

(continued from previous page)

```

    # (same structure as the input)

    plot_data1_out = alligator.plot_plugin_data(
        Plot_Name = 'Scaled Sum of Plots',
        X_Array = x1,
        Y_Array = sumy
    )
    plot_data2_out = alligator.plot_plugin_data(
        Plot_Name = 'Scaled Difference of Plots',
        X_Array = x1,
        Y_Array = diffy
    )
    plots_out = [plot_data1_out, plot_data2_out]
    message = 'Scaled Sum of Selected Plots (scaling factor: ' + \
        str(k) + ', '+name1 + ', '+name2+') and Phasor Frequency Update'
    graph_data_out = alligator.graph_plugin_data(
        Graph_Name = graph_name,
        Plots = plots_out,
        Reference_Decay = alligator.empty_plot)

    # Finally, we can send back information on the function outcome
    # and can also set AlliGator Parameters
    # all this packaged in a dictionary, converted to json and
    # appended to the (generally) empty string list
    # addtl_params_out_json_list

    info_out_dict = {
        "Notebook Message" : message,
        "Exception Type" : exception_type,
        "Exception Message" : exception_message,
        "AlliGator:Phasor Frequency" : f
    }

    # conversion to JSON string and string is appended to the incoming
    # addtl_params_out_json_list

    addtl_params_out_json_list.append(json.dumps(info_out_dict))

    # return results to AlliGator

    return(graph_data_out)

```

1.7.3 AlliGator Python Plugins API v 0.1

AlliGator versions: 0.63-0.75

parameter types

The following input parameter types are supported:

Type	Example	Description
U8	1	unsigned byte
U16	1	unsigned short integer
U32	1	unsigned long integer
I8	-1	signed byte
I16	-1	signed short integer
I32	-1	signed long integer
float32	1.0	single-precision floating point number
float64	1.0	double-precision floating point number
string	a string	string
path	C:\Desktop\test.txt	file or folder path
boolean	True	boolean: True or False
AlliGator	Phasor Frequency	any exposed AlliGator parameter (*)

(*) Requested AlliGator parameters are provided automatically to the plugin function. to obtain a list of current exposed internal AlliGator parameters, use the *Send* button in the **Settings:Plugins** panel. This will send a copy of that list to the Clipboard, together with the current parameter values.

A list of exposed parameters with their default values can be found [here](#)

Check the *Parameter Names only* checkbox in the **Settings:Plugins** panel to get a list of exposed internal AlliGator parameters without their current value.

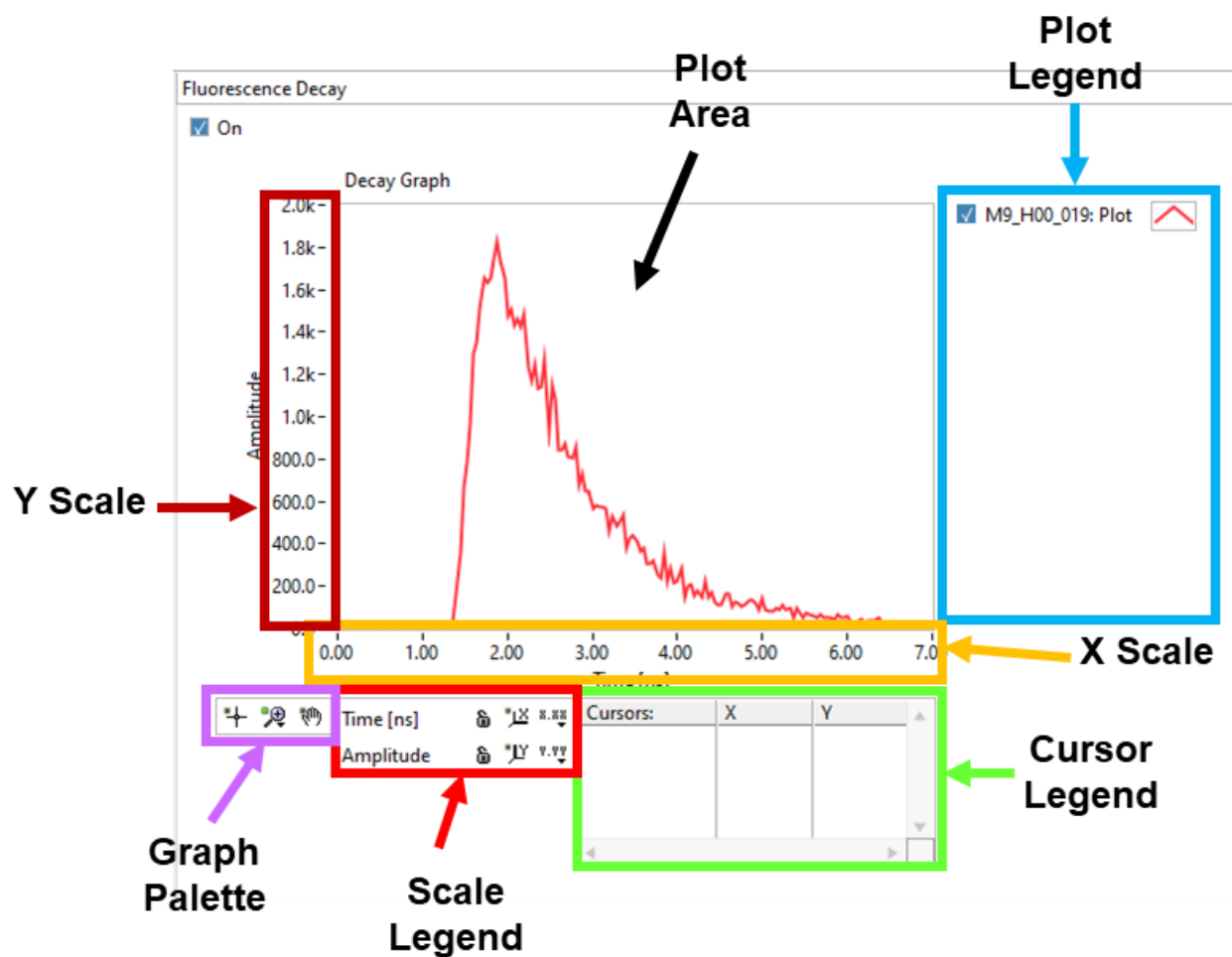
1.7.4 Graph Object

Multiplot graph objects are ubiquitous in AlliGator and are packed with many functionalities. Not all functions are available in all graphs (it wouldn't make sense to fit a decay model to an intensity histogram, for instance), therefore custom function descriptions will be found in page manuals dealing with each specific graph object. The following will hold for most graphs, with some exceptions.

Graph Object Anatomy

A graph object is comprised of:

- a *plot area*
- a *plot legend*
- two or more *graph scales*
- a *graph palette*
- a *scale legend*
- a *cursor legend*



Plot Area

The *plot area* is the rectangular region in which plots are displayed. In the example shown above, a single plot is displayed, but in general, more than one plot will be present. The list of available plots is found on the right in the *plot legend*.

The appearance of the plot area can be modified in the following manner:

- by modifying the range of displayed values (using the X and Y *scales* directly, or using the *graph palette*)
- by showing or hiding grid lines (using the *graph custom menu* or the *scale legend*)

The different actions will be discussed in the respective object's subsections below.

Plot Legend

The *plot legend* lists the plots stored in memory. Plot names do not need to be different, but it is easier to distinguish them if they are, especially when the list is long. In that case, a scrollbar will appear on the righthand side of the legend. Move the scrollbar up and down using the mouse wheel to show the name of all plots. Alternatively, use the *Index Display* control above the legend to select the index of the plot at the top of the list (the first plot has index 0).

Long plot names will not show up fully in the legend, but hovering over a plot name with the mouse will reveal a 'tooltip' with the full name of the plot.

Plot names can be edited by clicking in them.

Each plot's name is preceded by a checkbox, which defines whether or not the plot is actually displayed in the plot area. These checkboxes are also used as 'selection' checkboxes to perform custom actions on multiple plots at once, such as for instance saving the selected plots or changing their style. In other words, when performing an action on selected plots, the unselected plots are not visible in the graph.

The small image on the right of each plot name (*plot icon*) can be clicked on (right or left button) to give access to the standard plot menu.

Plot Menu



This menu allows (among other things) to specify:

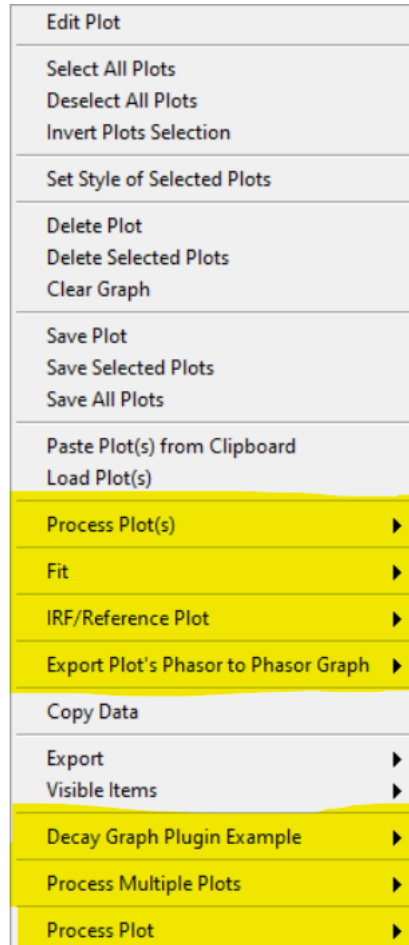
- the type of the plot (symbols, line, line + symbols, etc.)
- its color, line style, thickness and anti-aliasing
- the color and style of the symbols (if used)
- the scales (X and Y) associated with the plot (when there are more than one scale available per axis)

The **Export** options of this menu are less extended than those available through the *custom menu* described next.

Custom Graph Menu

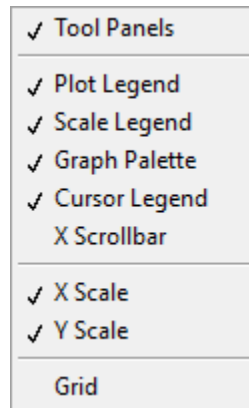
The following custom menu (accessible by right-clicking anywhere in the graph except for the plot icons and the *cursor table*) is specific of the *Decay Graph* in AlliGator, but many of its items are common to most graphs.

The most common are described next.



- Edit Plot: opens the Plot Editor window.
- Select All Plots does as it says.
- Deselect All Plots as well.
- Invert Plots Selection allows rapidly inverting the selected and deselected plots.
- Set Style of Selected Plots opens a *dialog window* to change the style of all selected plots at once.
- Delete, Delete Selected Plots and Delete All Plots or Clear Graph are self-explanatory (and irreversible)
- Save Plot, Save Selected Plots and Save All Plots allow saving plots as ASCII files (TAB separated columns of floating point numbers).
- Load Plot(s) opens a file dialog window with which one or more such ASCII files can be selected and loaded.
- Copy Data copies the Graph's image to the clipboard.

- **Export:** does the same thing as the **Export** submenu of the plot menu, that is, either export the selected plot data to the clipboard or to Excel
- **Visible Items**



The **Visible Items** menu allows rapidly hiding/showing all objects except for the *Plot Area* and *Plot Legend* using the **Tool Panels** item (useful for instance to copy/paste the graph image to the Notebook without the *Graph Palette*, *Scale Legend* and *Cursor Legend*), or different objects of the graph individually.

The **Grid** item of that menu allows rapidly showing/hiding a grid in the *Plot Area*, rather than using the individual axes menu in the *Scale Legend*.

Graph Scales

Most graphs have a single X (horizontal) and Y (vertical) scale, but some may have two or more, on either side of the graph (top or bottom for the X scales and left or right for the Y scales).

The visible bounds (min and max) of each scale can be edited by clicking on their respective values. Likewise, the scale title can be modified by clicking on it and editing it.

Further modification can be performed using the *Scale Legend*.

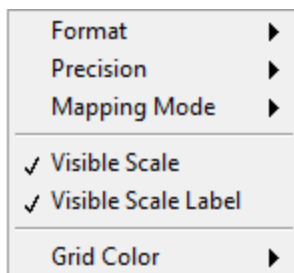
Scale Legend

The *Scale Legend* comprises the scale title (editable), an autoscale lock button, a single-autoscale button and a scale options menu.

The autoscale lock enables the graph to adjust its range automatically when a plot is added, so that all selected plots are visible. If this is not desirable, click the button to disable the autoscale option.

The single-autoscale button allows applying autoscale once only.

Finally, the scale options menu shown below provides option to format the scale tick labels, set the displayed precision, and choose between linear and logarithmic scaling. It also enables hiding a scale or only its labels and select the grid color.

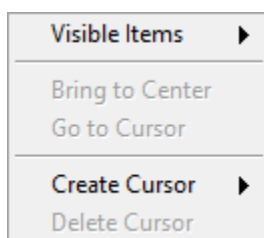


Graph Palette

The *Graph Palette* allows switching between tools to interact with the plot area:

- cross: cursor manipulation
- magnifying glass: zoom tools to adjust the plot area range(s)
- hand: panning of the plot area

Cursor Legend



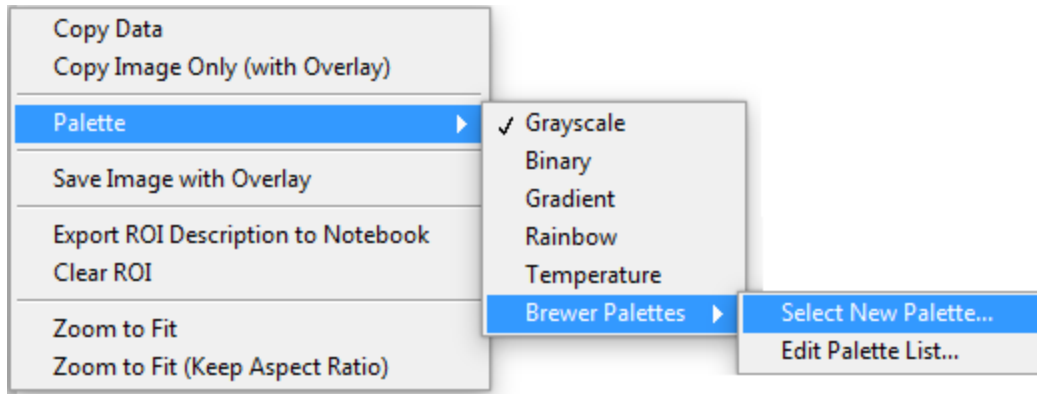
The *Cursor Legend* (or *Cursor Table*) shows the list of available cursors, to which individual cursors can be added (**Create Cursor**) or can be removed from (**Delete Cursor**).

The table also gives access to individual cursor properties (style, color, etc.) as well as associated plots, coordinates and plot value(s).

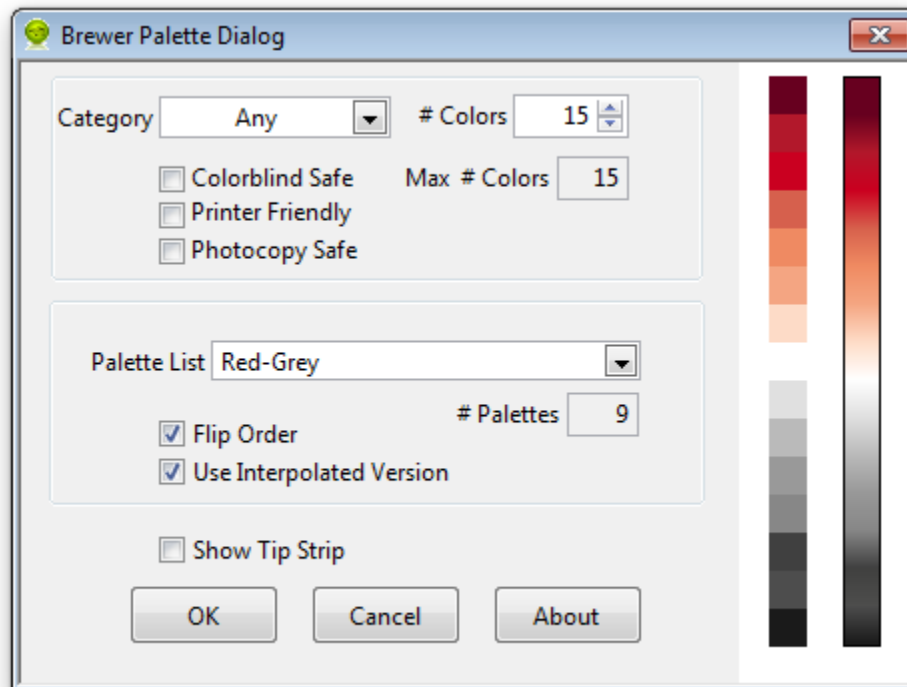
Finally, **Bring to Center** and **Go to Cursor** move the selected cursor to the center of the *Plot Area* or recenter the *Plot Area* on the selected cursor. The former provides a convenient way to locate a cursor and then fine-tune its location.

1.7.5 Image Color Palette

Images can be displayed with different color palettes accessible via the right-click menu of these images:

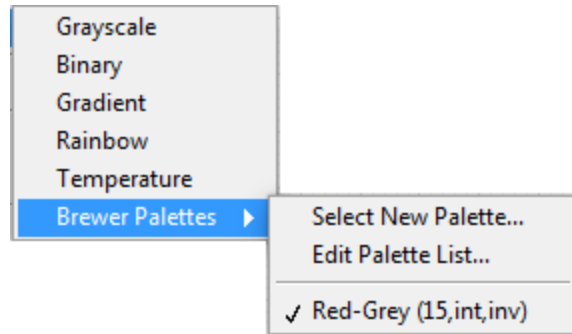


In addition to the built-in Grayscale, Binary, Gradient, Rainbow and Temperature palettes, the user has access to the palettes created by Cynthia Brewer (<https://colorbrewer2.org>), by choosing the Select New Palette... option. This opens up a dialog box offering a choice of palettes sorted by category (Any, Sequential, Diverging, Qualitative), which can be further constrained by selecting options such as Colorblind Safe, Printer Friendly or Photocopy Safe or number of colors (# Colors, from 3 to 15):



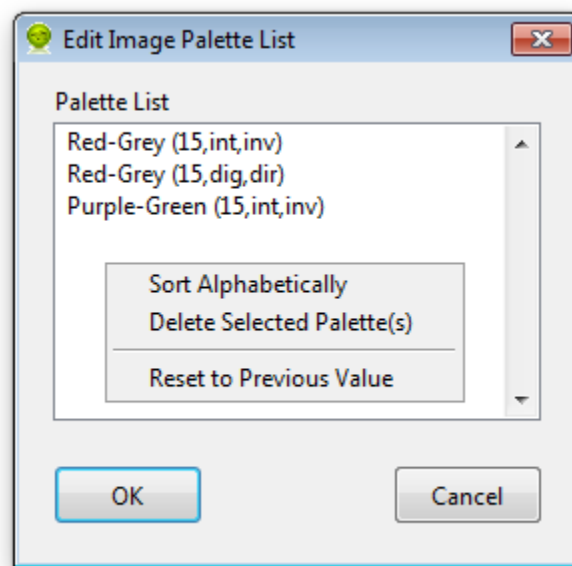
Note that when categories or options are selected, the maximum number of colors available for the remaining palettes might be reduced. In this case, the number of colors requested by the user will be limited by this maximum. The palette itself is chosen using the Palette List pull-down menu. The list is adjusted depending on the chosen category and options, as well as number of requested colors. The colors comprising the palette are displayed on the left side of the dialog box (squares stacked on top of each other), the first color being the bottom one (*). To inverse the order of the list, check the Flip Order checkbox. To use a color palette with 256 colors, check the Use Interpolated Version checkbox. The interpolated color palette appears to the left.

When you are satisfied with your choice, press the OK button. The selected palette is applied to the image and is added to the list of palettes in the right-click menu:



The palette name includes the number of colors and whether it is interpolated (int or dig) or inverted (inv or dir).

The list of Brewer palettes accessible via the right-click menu can be modified using the Edit Palette List... menu item. This menu opens a dialog box indicating the current palettes in the list:

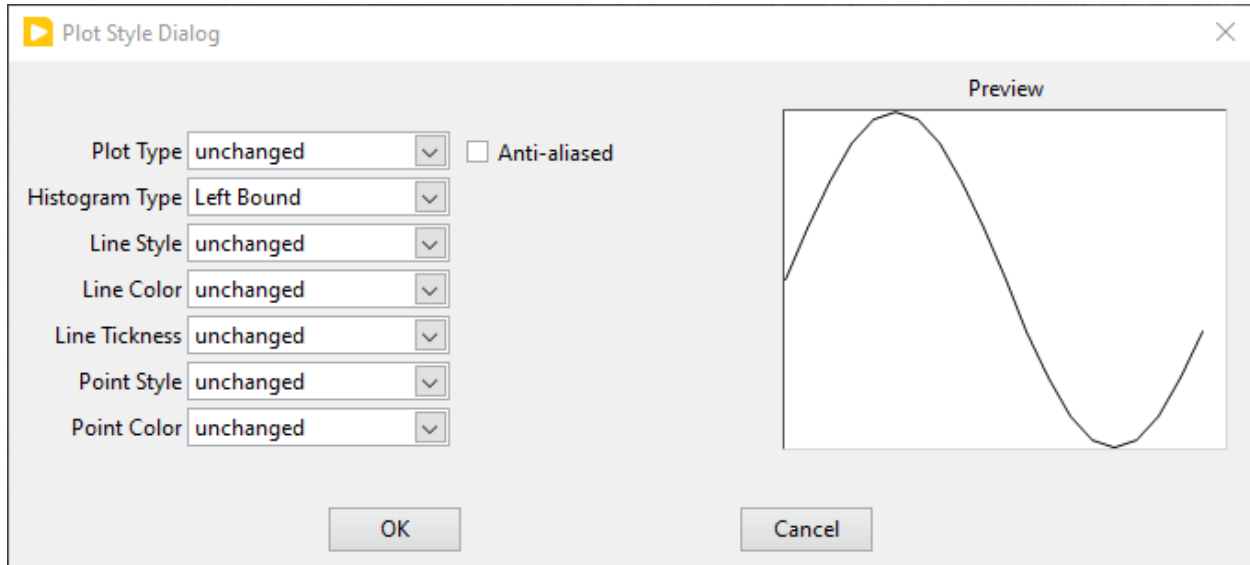


Items in this list can be manipulated by selecting and dragging them or by using the right-click menu items: Sort Alphabetically, Delete Selected Palette(s) and Reset to Previous Value. To confirm these modification, press OK, or Cancel to leave the list untouched.

(*) to change the background color, click next to the palettes to show a color picker.

1.7.6 Selected Plot Style Window

The **Set Style of Selected Plots** custom graph menu item opens up the following **Plot Style Dialog** window:

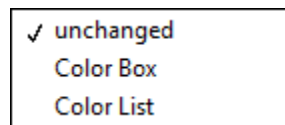


The *Preview* panel on the right will illustrate what plot style has been defined using the list of controls on the left.

These controls try to reproduce the options normally available for *individual* plots using the right-click menu associated with each plot's icon in the *Graph*, with some modifications. Importantly, the selected style(s) will be applied to *all selected plots*.

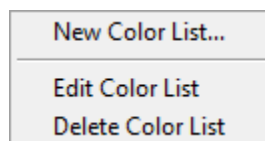
Notice that most of the selected style options are **unchanged** in the above snapshot. This means that the corresponding component of the plots' styles will not be changed. However, the **Histogram Type** parameter has been chosen to be **Left Bound**. The result of pressing OK will be that all selected plots which are "histograms" will be left-justified, while other plots will remain unchanged.

Most of the options are self-explanatory, but the two **Color** options (**Line Color** and **Point Color**) offer the flexibility needed for series of plots:



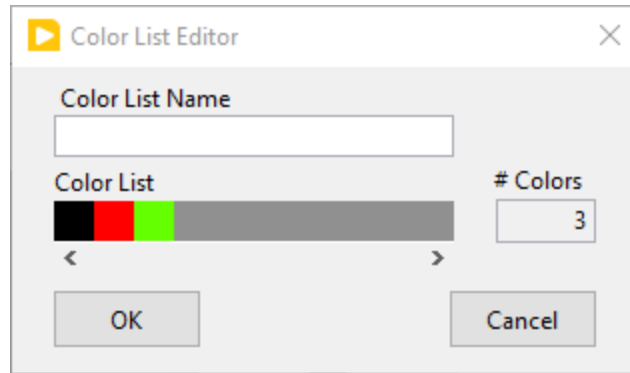
The **Color Box** option will show a color box next to the option, allowing to apply the selected color to all plots (either the line color or the point color, or both).

Selecting the **Color List** option will show a pull-down list next to the option. If it contains some color list names, one of them can be selected. Otherwise, right-clicking on the list will show the following menu:



Edit Color List and **Delete Color List** are self-explanatory.

Selecting **New Color List...** opens the following dialog window:



Clicking on individual *Color Boxes* in the linear array visible in the center, allows defining a series of colors. This series will be repeatedly applied to the selected plots when leaving the *Plot Style Editor*. It is recommended to give each user-defined color list a distinctive name for future use (these lists are saved with AlliGator Settings, therefore they are available from one run to the next).

To store the edited list, click OK and then OK in the *Plot Style Editor* window to apply the selected styles.

To leave the selected plots untouched, simply click on Cancel or close the *Plot Style Editor* window.

1.8 AlliGator Tutorials

The following is a list of video tutorials created by Jason Smith at RPI:

- [Data upload and decay visualization](#)

1.9 AlliGator Shortcuts

The following keyboard shortcuts are available when the main AlliGator window is in focus:

Shortcut	Action
Ctrl+A	ROI Analysis
Ctrl+Shift+A	All ROIs Analysis
Ctrl+B	Load B&H .sdt FLIM File
Ctrl+D	Fluorescence Decay Statistics
Ctrl+E	Open/Show Settings
Ctrl+Shift+E	Export Settings to Notebook
Ctrl+F	Save Settings
Ctrl+Shift+F	Load Settings
Ctrl+F1	Open SwissSPAD 1 Dataset
Ctrl+H	Show/Hide Context Help
Ctrl+I	Show Dataset Information Window
Ctrl+L	Load Image Folder
Ctrl+Shift+L	Load Image Folder Series
Ctrl+O	Open HDF5 Dataset
Ctrl+Shift+O	Open HDF5 Dataset Series
Ctrl+N	Open/Show Notebook
Ctrl+P	Playback Time-Series
Ctrl+Shift+P	Loop Time-Series
Ctrl+Q	Quit AlliGator
Ctrl+R	Reload Dataset
Ctrl+Shift+R	Open Source Image ROI Manager
Ctrl+Shift+S	Save as HDF5 Dataset
Ctrl+T	Time-Series Phasor Analysis
Ctrl+Shift+T	Time-Series Decay Fit Analysis

1.10 AlliGator Installation

1.10.1 System Requirements

This software will only run on Microsoft Windows 10 64 bit. Some of the functions used by the software support parallelism, therefore a multiprocessor and/or multicore computer is recommended (this statement is not as dated as it sounds, as virtual machines offer the choice of core number). Since time-gated image datasets tend to be large, and many analysis steps involve processing all time-gate images, it is preferable to have as much RAM as possible available. 16 GB or more are recommended.

1.10.2 Download the installer(s)

3 pieces of software need to be present on your computer for AlliGator to run properly:

- AlliGator
- LabVIEW Run-time Engine (RTE)
- Vision Development Module (VDM) RTE

Follow the instructions below to make sure you install the correct versions of each. Do not hesitate to post a support request on the [AlliGator Support Google Group](#) to ask for help or advice. You only need to install the LabVIEW and VDM RTE once.

AlliGator Installer

Find and download the latest installer zip archive in the `installer/latest` folder of the AlliGator GitHub repository.

Installers for previous versions can be found in the `installer` folder and can be run to overwrite a newer version containing a bug or to replicate published analysis performed with an older version.

Note that older AlliGator versions may require older LabVIEW and VDM RTE, as explained in the *AlliGator and LabVIEW versions* section below.

LabVIEW and Vision Development Module RTE

The AlliGator installer only installs the software and its dependent files, but won't result in a functional executable unless you have the LabVIEW 2021 SP1 64 bit Run-Time Engine (RTE) installed on your machine, as well as the Vision Development Module (VDM) 2021 Run-time Engine.

The LabVIEW RTE installer is available for free from National Instruments at [<https://www.ni.com/en/support/downloads/software-products/download.labview-runtime.html#443841>](https://www.ni.com/en/support/downloads/software-products/download.labview-runtime.html#443841)

Although the VDM RTE is free to download at [<https://www.ni.com/en/support/downloads/software-products/download.vision-development-module-runtime.html#409837>](https://www.ni.com/en/support/downloads/software-products/download.vision-development-module-runtime.html#409837), it requires a license to be used (asked for during installation). As explained [here](#) a trial license can be requested from National Instruments/Emerson, but it will expire after a one week trial period. If you have a license for a different version of VDM, it will work. As of 2/2024, an academic license costs \$407.40 (\$582 for non-academic users).

Installation Instructions

Once the LabVIEW RTE and then the VDM RTE are installed (this only needs to be done once), run the `install.exe` file in the AlliGator installer zip file (there is no need to unzip the installer). The default installation folder is in the logged Windows user's personal folder, therefore no Administrator right is needed.

There is no need to uninstall a previous version to install a new one (which can be done from within AlliGator using the `Check for Update` item of the Help menu).

AlliGator and LabVIEW Versions

AlliGator is developed using the graphical programming language LabVIEW and uses the Vision Development Module (VDM) toolkit for some image handling and analysis functions. The software therefore requires the LabVIEW runtime engine (RTE) and the VDM RTE corresponding to the version it was developed with.

Note that for AlliGator versions anterior to 0.80, version 1.8.18 of the HDF5 library needs to be installed separately, as explained in section *HDF5 library installation* below.

Over the years, different LabVIEW versions have been used and therefore, different versions of AlliGator require different versions of the LabVIEW and VDM RTE, as explained below.

- **LabVIEW 2021 SP1 64 bit for version 0.51 and above**

- LabVIEW RTE: [<https://www.ni.com/en/support/downloads/software-products/download.labview-runtime.html#443841>](https://www.ni.com/en/support/downloads/software-products/download.labview-runtime.html#443841)
- VDM RTE: [<https://www.ni.com/en/support/downloads/software-products/download.vision-development-module-runtime.html#409837>](https://www.ni.com/en/support/downloads/software-products/download.vision-development-module-runtime.html#409837)

- **LabVIEW 2019 SP1 64 bit for version 0.21 to 0.50**

- LabVIEW RTE: <https://www.ni.com/en/support/downloads/software-products/download.labview.html#348045>

- VDM RTE: <<https://www.ni.com/en-us/support/downloads/software-products/download.vision-development-module.html#329029>>
- **LabVIEW 2018 SP1 64 bit for version 0.15 to 0.20.4**
 - LabVIEW RTE: <<https://www.ni.com/en-us/support/downloads/software-products/download.labview.html#309628>>
 - VDM RTE: <<https://www.ni.com/en-us/support/downloads/software-products/download.vision-development-module.html#306495>>
- **LabVIEW 2017 SP1 64 bit for version 0.9.11 to 0.15**
 - LabVIEW RTE: <<https://www.ni.com/en-us/support/downloads/software-products/download.labview.html#310821>>
 - VDM RTE: <<https://www.ni.com/en-us/support/downloads/software-products/download.vision-development-module.html#306485>>
- **LabVIEW 2016 64 bit for versions prior to 0.9.11**
 - LabVIEW RTE: <<https://www.ni.com/en-us/support/downloads/software-products/download.labview.html#310817>>
 - LabVIEW VDM RTE: <<https://www.ni.com/en-us/support/downloads/software-products/download.vision-development-module.html#306484>>

Several versions of the RTE can be installed concurrently on the same machine, but since they are hefty, it is possible to uninstall the unused versions to save space.

Uninstalling AlliGator does not uninstall the corresponding RTE version, therefore it is only necessary to install the required RTE once.

Note that if you have the correct *LabVIEW development environment* installed on your machine, there is no need to install the RTE again, as it will already be available.

HDF5 library installation

In order to be able to save and load HDF5 files introduced in version 0.16, HDF5 library version 1.8.18 needs to be installed *for AlliGator versions anterior to version 0.80*. AlliGator 0.80 and above include the HDF5 library (version 2.15.0.149) in the installer, therefore this step can be ignored.

The free HDF5 library installer archive can be found here: <<https://support.hdfgroup.org/ftp/HDF5/releases/hdf5-1.8/hdf5-1.8.18/bin/windows/hdf5-1.8.18-win64-vs2013-shared.zip>>

1.11 AlliGator Version History

Note: version numbers followed by an asterisk have not been released.

1.11.1 0.86 (2024-04-11)

- **Bug Fixes**
 - Fixed empty phasor added to Phasor Graph (when an invalid decay was generated)
 - Fixed occasional slow ROI Decay Analysis
 - Fixed default folder in “Save Phasor Ratio Reference(s)” dialog
 - Fixed Default Max Percentile in Fit Options to 0
- **New Features**
 - Refreshed icon
 - Added “Set Reference n to User-entered Lifetime” menu items to Phasor Graph & Phasor Plot
 - Added Delete ROIs based on Statistics option in the Source Image ROIs menu
 - Fixed and updated Image Auto-Contrast option
 - Added Lower and Upper Percentile Targets in the Image Histogram Panel
 - Changed default name of Multiple ROIs file to include the name of the dataset

1.11.2 0.85.1 (2024-04-04)

- **Bug Fix**
 - Fixed slow single-pixel ROI to Decay extraction, speeding up related pixel-based analyses

1.11.3 0.85 (2024-03-31)

- **Bug Fixes**
 - Fixed bug in Image Histogram busy icon
 - Fixed non-functional Intensity Time Trace computation when processing a Series
 - Fixed unresponsive “Use File Time Stamp” checkbox in Intensity Time Trace panel
 - Fixed bug that would not load a common background correction in a Series Sum calculation (only the first dataset in the series would be corrected)
 - Fixed bug that required the user to select the gate channel of all SS3 datasets in a series
 - Minor cosmetic fixes
- **New Features**
 - Added (Rotated) Rectangle Image Profile (aka “thick” line image profile)

1.11.4 0.84 (2024-03-08)

- **Bug Fixes**

- Fixed behavior of “Use File Timestamp” in Intensity Time Trace panel and Settings
- Fixed bug in transcription of date/time information for PI Imaging dataset in the Dataset
- Fixed broken loading of .phplot files in Phasor Graph

1.11.5 0.83 (2024-02-27)

- **Bug Fixes**

- Fixed broken Phasor Ratio Map Palette selection menu
- Fixed bug that prevented highlighting phasor ROI in Source Image
- Fixed broken loading of Image Folder without metadata (which would erase stored settings)
- Fixed parsing of version history
- Fixed bug in Gate Image Folder loading which would fail to load non-PI Imaging datasets

- **New Features**

- Speed up of Source Image zoom responsiveness when an image overlay is applied
- Added options to blend overlay color to background image in Source Image and Phasor Plot Settings panels
- Added label style options to the Phasor Plot Settings tab

1.11.6 0.82.1 (2024-02-14)

- **Bug Fix**

- Fixed bug in Gate Image Folder loading which would fail to load non-PI Imaging datasets

1.11.7 0.82 (2024-02-07)

- **Bug Fixes**

- Fixed a bug with Mouse Move events in Image Display & Phasor Plot introduced in v0.81 which broke a number of UI functionalities
- Fixed a bug that could set the wrong frequency for subsequent calculation when manually exporting a plot to the Phasor Graph
- Improved responsivity to Image Histogram Cursor Move events

- **New Features**

- Added “Shot Noise Influence on Average Lifetime” to the Analysis>>Tools menu
- Deprecated “Simple Search” algorithm when using the Phasor Calibration Map option
- Added Dialog allowing to use the frequency of the calibration phasor (or phasor map/phasor series) if it is different from the current phasor frequency

1.11.8 0.81 (2024-02-03)

- **Bug Fixes**
 - Fixed missing “Slope” metric plot and data in IRF extraction
 - Improved Mouse Move event handling in UI
- **New Features**
 - Added “Draw Reference Segment” checkbox in Settings>>Phasor Plot
 - Added “Dataset Scaling Factor” to Settings>>Data Information
 - Added workaround to the bit padding implemented in SPAD512 software
 - Skipping “Gate Steps” parameter in PI Imaging PNG metadata, as it is erroneous in version 1.43 and prior

1.11.9 0.80 (2024-01-16)

- **Bug Fixes**
 - Fixed bug whereby when opening a child window, clicking into the Image panel would result in a black background
 - Fixed unresponsive “Check for update” menu item
 - Fixed plot transposition when computing Parameter 2 vs Parameter 1 in Phasor Graph
 - Fixed bug in Sum All FLI Datasets in Series preventing some sums to be computed
 - Fixed Notebook reopening with the default Notebook name even when saved (and sometimes with empty content)
- **New Features**
 - upgraded to v 2.15.0.149 of h5labview2 library (which should in principle dispense from having to install the HDF5 library separately)
 - AlliGator user event handling refactoring to reduce lag following Mouse Move events
 - Added Load .bin FLI Dataset & Dataset Series (Piccolo) to File menu
 - Renaming of File loading menu item for clarification
 - Changed menu location of Multi-ROI Single-Pixel scripts, now located in the Analysis menu
 - Added ROI description in Time Series Analysis Notebook output if the ROI is not one of the stored ones
 - Added progress bar when file loading

1.11.10 0.79.2 (2024-01-05)**• Bug Fixes**

- Fixed “forgotten” calibration when refreshing the Phasor Plot
- Fixed mix-up between two SDV options in Multiple ROI Phasor Analysis (fast, non-interactive)
- Fixed a bug that would return an error when using 1/D as the phasor frequency to use when exporting a decay’s phasor to the Phasor Graph
- Fixed “Export JSON string to Clipboard” in Settings>> Phasor Frequency controls.

1.11.11 0.79.1 (2024-01-04)**• Bug Fixes**

- Fixed bug in calculation of calibrated phasor when SDV calculation option is checked [Analysis>>FLI Dataset>>Multiple ROIs>>Phasor Analysis (fast,non-interactive)]
- Fixed bug in calculation of calibrated phasor when Use Estimated Average Phasor option is checked [Analysis>>FLI Dataset>>Multiple ROIs>>Phasor Analysis (fast,non-interactive)]

1.11.12 0.79 (2024-01-04)**• Bug Fixes**

- Fixed downloading and installing of new version
- Fixed truncation of Folder Name when dealing with Gate Image folder
- Fixed bogus message when loading a new file with the same background subtraction parameters as the previous one
- Fixed sluggish Phasor Plot Reference user-controlled motion
- Fixed long “Highlight Image ROI in Phasor Plot” step when loading new dataset and a full-frame ROI is drawn
- Phasor Ratio Color Scale is now updated according to “Color Map Type”
- Fixed erroneous display of Average Lifetime range in Settings>>Phasor Plot
- Fixed Phasor Ratio/Average Lifetime overlay responsiveness
- Fixed missing internal update of Phasor Ratio/Average Lifetime Display Range parameter
- Fixed long-standing bug in Phasor Graph where the internal state wasn’t updated properly when one or more plots were deleted from the graph (clearing the graph was behaving as expected)

• New Features

- Added “Transpose Plot”, “Merge Selected Plots”, “Plot Histogram” to Decay Graph
- Added support for “Date taken” and “Time taken” of SPAD512^2 files
- Added support for Background Series
- Added dialog to remind the user that references are needed to compute a phasor ratio or average lifetime overlay
- Added “Close” sub-menu to the File menu (e.g. Closing a Dataset Series before switching to an individual dataset is recommended to avoid future data confusion)

1.11.13 0.78 (2023-12-22)

- **Bug Fixes**

- Fixed typos in GUI
- Fixed bug in Fit Parameters Options which ignored the “Numerically Estimated” option unless the array of Guess Parameters was empty
- Fixed bug in pixel-to-pixel phasor map calibration
- Fixed bug preventing loading old phasor calibrations and phasor calibration maps
- Fixed bug that would prevent reloading of dataset when the background dataset type is different from that of the dataset type (e.g. HDF5 vs Gate Folder)

- **New Features**

- Added “Move Flattened Series to Folder Series” menu item in File>>Load>>FLI Dataset Series menu (to deal with SPAD512 flattened output)
- Moved Analysis>>FLI Dataset>>Multiple ROI Analysis>>Multi-ROI Single-Pixel NLSF and Phasor Analysis to Scripts menu
- Removed “Pixel-to-Pixel” calibration option (now automatically applied when a full image calibration map is stored)

1.11.14 0.77 (2023-12-16)

- **Bug Fixes**

- Fixed online version check (no message if installed version is the same as latest release)
- Fixed bug where deactivating Phasor Calibration Map did not result in a Phasor Plot update
- Fixed missing shortcut menu items in Source Image ROI Manager
- Fixed Complementary ROI calculation

- **New Features**

- Added “Use Estimated Average Phasor” checkbox as part of the phasor standard deviation option in Settings:Phasor Graph
- Added “Pixel-to-Pixel” Phasor Map Calibration for significant speed up in cases where the Calibration Map is computed for all pixels in the dataset
- Implementation of parameter SDVs in “Export Phasor Plot Add'l Data” in Phasor Graph
- Added support for greyscale RGB images
- Changed ROI connectivity to 4 in ROI Mask to ROI conversion
- Support for amplitude- or intensity-averaged lifetime output in FLI Dataset Series>>Series Phasor Analysis
- Support for Phasor Ratio and Averaged Lifetime SDV in FLI Dataset Series>>Series Phasor Analysis
- Consolidated “All ROIs NLSF Analysis>>Non-Interactive (Fast, Common IRF)” and “... (Fast, Individual IRF)” menu items, the option being set by the “Settings>>Fluorescence Decay>>Fit Options>>Use Local IRF” parameter

1.11.15 0.76 (2023-12-03)

- **Bug Fixes**

- Change to Phasor Calibration Options, Gate Name, Gate Arithmetic, etc. now flags Phasor Plot for update
- Fixed Harmonic Frequency context menu in Settings:SEPL panel
- Fixed bug in Amplitude Phasor Ratio SDV calculation
- Set # processors used during initialization

- **New Features**

- Local Decay Graph window now displays decay for any kind of ROI (not just single-pixel ROI)
- Deprecated Source Image:Average Dataset Sum, Source Image:Use Phasor Plot Binning settings
- Shortened list of information sent to the Notebook in verbose mode when a new Phasor Plot is calculated
- Added “Parameter 2 vs Parameter 1 Scatter Plot” in Phasor Graph
- Removed “Phase Lifetime Trajectory” as being superseded by “Parameter 2 vs Parameter 1 Scatter Plot”
- Removed “Phase Lifetime vs Parameter” scatter plot sub-menu as being superseded by “Parameter 2 vs Parameter 1 Scatter Plot”
- Added “Use Estimated Average Phasor” checkbox as part of the phasor standard deviation option in Settings:Phasor Graph

1.11.16 0.75.2 (2023-11-26)

- **Bug Fixes**

- Fixed update of Fit Options:Laser Period when loading a new file and Use Data Information Laser Period = TRUE
- Fixed last version detection

1.11.17 0.75.1 (2023-11-22)

- **Bug Fixes**

- Fixed “Use Data Information Laser Period” not updating the laser period used for fitting

- **New Features**

- Changed approach to compute offset error estimation: now computed at “optimal” χ^2 obtained by parabolic estimation, while the actual returned offset parameter is that of the best fit.

1.11.18 0.75 (2023-11-19)

- **Bug Fixes**

- Amplitude- or Intensity-averaged lifetime shown on Phasor Graph now matches that displayed in the Notebook (when the phasor SDV option is not selected, it is the value computed from the actual decay, while when that option is selected, it is the average of all replicas)
- Fixed a bug in “All ROIs NLSF Analysis>>Non-Interactive (Fast, Individual IRF)” that resulted in incorrect fit results

- **New Features**

- Output of both amplitude- & intensity-averaged lifetime and phasor ratio in average phasor calculation (Phasor Graph right-click menu)
- Support for “Polygon” ROI Image Profile (the ROIs created via “Create ROIs with Intensity above Min” are Polygons)
- Deprecation of “second pass” attempt to improve fit in “All ROIs NLSF Analysis>>Non-Interactive (Fast, Individual IRF)”

1.11.19 0.74 (2023-11-12)

- **Bug Fixes**

- Fixed “Load/Save Reference(s)” menu item in Phasor Plot
- Locked panel splitters
- Fixed new version check
- Fixed SEPL parameter updates in Settings
- Fixed missing gate exposure, duration and integration time information in saved HDF5 FLI dataset
- Fixed missing Settings>>Data Information>>Gate Separation update
- Removed “Interpolate Plot” (not functional)

- **New Features**

- Added Option in Decay Graph to Export a plot’s phasor using the natural frequency 1/D
- Added option to ignore differences in phasor frequencies in Phasor Graph >> Phasor Multiplication
- Changed Phasor Plot calculation message to optional (only visible when “Verbose Mode” is on)
- Set new Default Cursor Style to a more contrasted one
- Added “Pair Selected Plots” menu item in Lifetime & Other Parameters graph to combine the Y axes of two plots into a single Y2 vs Y1 plot
- Added option to enforce that the laser period used in fitting is identical to that of the Data Information
- Added IRF Deconvolution option to use the IRF tail slope as selection criterion
- Replaced outliers by zero in deconvolved IRFs
- Simplified Image Histogram Options (default: 256 bins)
- Added “Process Single Plot/Selected Plots/All Plots” option in Decay Graph
- Implemented Multiple Plot processing for: 1-normalization, [0-1]-normalization, rebin, cumulative function, denoise, extrapolate, fold, shift, smoothen, smoothen tail, straighten

- Rearranged/renamed Decay Graph shortcut menu
- Implemented calculation of Phasor SDV, Phase & Modulus Lifetime SDV, Phasor Ratio SDV, Amplitude- & Intensity-Averaged Lifetime SDV
- Added indicators for the derived quantities in the Phasor Graph panel
- Added “Convert (All) ROI(s) to Mask Image” menu item to Source Image
- Added support for PI Imaging PNG Metadata information

1.11.20 0.73 (2023-09-18)

- **Bug Fixes**

- Fixed a bug in High Count Image Pixel Rejection which would reject all pixels when the “Reject High Count Pixels” option was selected
- Updates Decay Fit Parameters Map when fit is completed
- Fixed Dataset Image update when loading new file
- Fixed bug using different source for Laser Period parameter in NLSF fit and fit display

- **New Features**

- Added dialog during background subtraction, to allow using files datasets with different properties (e.g. laser period)
- Now loading SSX Detector Information from HDF5 and adding it to the metadata
- Added Compute & Plot All ROIs Statistic
- Implemented plot of Decay Fit Parameter vs Intensity
- Changed Settings>>Fluorescence Decay>>Fit Parameters>>Guess Parameter Options to drop-down list instead of series of checkboxes
- Added Min of All Gates displayed image option
- Added Decay Preprocessing Options & Reference Decay in the data saved with a Fit Parameter Map file
- Implemented “Merge Selected Plots” and “Sliced Mean & SDV Plots” in Lifetime & Other Parameters Graph
- Deprecated Fit Status Window (for performance improvements)
- Added shortcut menu “Use Data Information >> Laser Period” in Fit Options

1.11.21 0.72 (2023-08-10)

- **Bug Fixes**

- Fixed bug in histogram option preventing IRF square gate fits to work
- Fixed bugs in ROI Statistics which would fail when encountering Inf or NaN or Negative values
- Fixed bug in faster Hot Pixel Removal code
- Fixed Hot Pixel Removal options Notebook message
- Fixed truncated accumulated file name when the folder name contained a period

- Prevented multiline input in Settings>>Plugins>>Python Version
- **New Features**
 - Added “Save Image as Hot Pixel Mask” to Source Image context menu
 - Added “Use Hot Pixel Mask” and “Hot Pixel Mask Image” to Settings>Source Image Hot Pixel Removal Options
 - Processing of border of images when removing hot pixels
 - Reordered FLI Dataset Series actions
 - Changed “# Bins” Image Histogram single option to full-fledged “Image Histogram Options”
 - Changed “INT-normalized” Channel Arithmetic to $G2/INT * <INT>$
 - Removed useless Channel Arithmetic Combinations and added $(1-G2/INT) * <INT>$
 - Added Auto-Adjust Contrast to saved Settings
 - Minor updates to Decay Graph context menu
 - Added “Mask Image” menu item to File>>Save menu
 - Changed menu items from “Open” to “Load”

1.11.22 0.71 (2023-06-26)

- **Bug Fixes**
 - Removed duplicate vertices in Freehand Tool ROIs
 - Fixed cosmetic glitch in Phasor Plot display of local phasor information
 - Fixed Histogram update when changing Gate Slider position
 - Fixed bug in Histogram calculation when the first array element is NaN or Inf
 - Fixed glitch in Gate Selection Dialog which sometimes opened with a “-1” selection
 - Fixed bug improperly applying pile-up correction to Background Dataset
 - Fixed bug that would overwrite Gate Selection when using background correction
 - Skip Gate Selection Dialog when a single gate type is available
- **New Features**
 - Added Average Lifetime Profile to Image Profile Window Plots (when the Average Lifetime is selected as Overlay Type in the Settings>>Phasor Plot panel)
 - Added “ROI Description Max Characters” to Settings>>Miscellaneous
 - Added support of Drag & Drop loading of Phasor Ratio References
 - Added “Auto-Adjust Contrast” option in Image Histogram panel
 - Optimized noise level in Extrapolate Decay (only applies to decays that are not normalized)
 - Added “Overlay Analyzed ROI Center” option in Settings->Source Image (Default: False)
 - Changed Phasor Plot Destination (in Phasor Algebra tools) to enum (from 3-radio buttons control)
 - Added Save dual-channel FLI Dataset to HDF5
 - Added support for Dataset Channel switch without Dataset Reloading (when Channel Arithmetic = None)

1.11.23 0.70 (2023-05-30)

- **Bug Fixes**

- Support for Multi-ROIs Single-Pixel NLSF Fits with common IRF (handles parameters map properly although it only contains one IRF)
- Added working option to specify a User-Defined separator in Load Plot(s) Dialog
- Fixed bug in Load Plot(s) dialog which would not show the abscissa options on first call
- Fixed bug in Settings which used stale values of Decay Shift Parameters
- Fixed round-off error preventing a zero offset to be reached when letting the offset parameter float
- Fixed NaN parameter uncertainty when one of the parameter gradients equals 0
- Fixed Local Decay Window not quitting properly when AlliGator closes

- **New Features**

- Added Decay Graph “Plot Convolution” menu item
- Clean-up and refactoring of convolution code (using FFT)
- Minor edits to Phasor Math Dialog Window
- Modified Sequence of actions in Decay Fit (important change!)
- Implemented Fixed Parameters Multi-Exponential Decay Fit (vs “constrained in 0-size interval”)
- Implemental Analytical Computation of Partial Derivatives for NLSF Fit
- Added optional “Use Legacy Fitting Approach” in Settings>>Decay Analysis>>Fit Options
- Switched to fitting the square root of lifetimes to enforce positivity
- Implemented new fitted offset approach (parallelized and uncertainty estimate)
- Changed criterion for selection of “Best of All” Fits to max R^2 instead of min χ^2 (since the χ^2 is computed without weights for an unweighted fit)
- Added parallelism in some multiROI scripts
- Changed Default Fit Options Parameters to lower values to improve convergence and fit speed
- Reverted GUI Layout to v0.67

1.11.24 0.69 (2023-04-17)

- **Bug Fixes**

- Corrected bug in 1D Histogram that affected Image Histogram (among other things)
- Fixed AlliGator sluggishness
- Fixed a bug that prevented loaded Plots from being fitted unless a script had been run
- Fixed a bug in Multiplot loading when each plot is comprised of two axes (no common axis)
- Fixed a bug making submenu items inactive in Decay Fit Parameters Map
- Fixed a bug preventing Gate Image Series Analysis (Fitting & Phasor)
- Fixed Drag & Drop of Plots into the Lifetime & Other Parameters Graph
- Fixed mangled Notebook output for IRF Deconvolution

- Associate correct axes to Parameter Plots in Lifetime & Other Parameters Graph

- **New Features**

- Improved Phasor Plot decorations (ticks, labels and references ROI)
- Added busy icon for additional tasks
- Added Source Image Overlays as option to the Erase button in Phasor Plot
- Modifications to support SwissSPAD Live File version 0.7
- Added support for Mask Images comprised of 0 and 255 only
- Added Multiplot Math item in the Plot Math submenu of the Decay Graph
- Added Background Correction & Gate Arithmetic to Dataset Series Sum
- Using a transparent color for Phasor ROI/Image ROI highlight erases the corresponding highlight
- Added IRF Deconvolution Metrics option in Settings
- Restored Series Phasor Analysis menu item
- Added Notebook message when single image is processed (Mask or White Light)
- Added optional White Light Image background subtraction
- Check that a ROI is selected before starting a Series analysis
- Grayed out Global Fit of Selected Plots in Decay Graph
- Added display of phasor information when moving reference in Phasor Plot

1.11.25 0.68.1

- **Bug Fixes**

- Loading ROIs(s) by drag & drop now returns a Notebook message
- Fixed Send All Settings to Notebook

- **New Features**

- Added warning message in Multiple ROIs Multi-Pixel script dialog to remind the user to define Settings for the Dataset (and/or IRF)
- Speed up of Phasor Plot calibration by Phasor Calibration Map in the Phasor Graph

1.11.26 0.68 (2023-03-05)

- **Bug Fixes**

- Fixed error returned if an image folder is missing a metadata file: no returns a warning only
- Fixed missing progress bar in Multiple ROIs Analysis>>All ROIs Decay Fit Non-Interactive (Fast)
- Fixed erroneous update of internal variables when a decay is invalid
- Fixed generation of empty plot, bogus phasor, etc. when a decay is invalid (i.e. empty)
- Corrected typos in Notebook messages
- Fixed bug erasing gate parameters when loading Gate Folder without metadata file
- Fixed bug in Image Histogram that would slow down calculation for large images

- Fixed a bug in ROI to individual pixel ROI conversion (skipped when the ROI is not a closed shape)
- Fixed a bug preventing output of Settings to the Notebook
- Fixed a bug in Phasor Calibration that would compute $\text{abs}(g)$ instead of g
- Fixed a bug that would erase all overlays each time the Source Image was updated

- **New Features**

- Added Phasor Graph output message
- Added Bin Number option in Histogram options (Settings>>Miscellaneous)
- Moved ‘Compute Optimal Calibration from Phasor References’ Phasor Graph menu item
- Changed Phasor Calibration Type ‘No’ to ‘None’
- Implemented Phasor Graph Algebra GUI
- Simplified internal computation of calibration
- Minor improvements to calibration map handling
- AlliGator Calibration menu checkmark validation at startup
- Added option to display Phasor Calibration Map ROI Centers on Source Image (Calibration>>Show Phasor Calibration Map ROI Centers)
- Faster Phasor Plot calibration
- Added Phasor Plot Calibration Algorithm option: “Simple Search” or “Min Distance”
- Better support of “Backup Calibration Option” to enable combining “Simple Search” Map Calibration for the calibrated ROI(s) and “Single Calibration” for the remainder of the image
- Cosmetic changes to AlliGator main window
- Improved Notebook message when loading calibration and after Phasor Plot calculation
- Added option to not store the Phasor Harmonics computed for each Phasor Plot

1.11.27 0.67 (2023-01-23)

- **Bug Fixes**

- Fixed Phasor Plot error thrown when no Phasor Ratio References are defined
- Fixed unresponsive “FLI Dataset Series” menu items
- Fixed bug preventing from deleting a Phasor Plot in the Phasor Graph
- Fixed a bug preventing highlighting Image Source ROIs in the Phasor Plot

- **New Features**

- Removed requirement of providing a dummy plot in Decay Graph Python Plugin if no plot is output
- Correspondingly, removed the requirement of an input parameters and/or output parameters section, if no such parameters is passed (for instance if the function’s output is simply a string output to the Notebook, or a file, etc.)
- Supports sending plots to other graphs than the Decay Graph (except the Phasor Graph)
- Changed example Python Plugins script file names to XXX_Example.py to make it clear they are only examples

- Added option to precede a Python Plugin idem menu by a separator, by starting the function's name with an underscore
- Better error reporting for Python Plugins
- Added “Process Multiple Plots” submenu to Decay Graph as a Python Script. First function: Linear Combination (Selected Plots).
- Added Abort button and Progress Bar during FLI Dataset saving
- Improved Notebook message after FLI Dataset saving
- Made AlliGator window resizeable to allow larger Source Image display
- Added New Decay plot message
- Drag & Drop of files (or folder of images) still works but the target is now the Source Image (not any random location in the main AlliGator window). This works for both FLI Datasets (or Dataset Series) and plots, ROIs, phasor plots, calibration, etc. All should be dragged and dropped in the Source Image. AlliGator will figure out what type of file they are and dump them in the appropriate object.

1.11.28 0.66 (2022-12-31)

- **Bug Fixes**

- Fixed a bug in Python Plugins.zipped archive preventing from updating outdated Plugins

- **New Features**

- Added Compute Average Lifetime to Decay Graph>>Process Plot menu
- Added Average Lifetime Options to Settings>>Fluorescence Decay>>Advanced Decay Analysis
- Added Positive constraint on extracted IRF, by replacing negative values Y by a normally distributed values in $N(0, \text{abs}(Y)/3)$
- Added option to use a smoothed metrics for Optimal IRF Extraction
- Added corresponding Savitzky-Golay parameters in Settings (Number of Side Points, Polynomial Order)
- Added “Create Complementary ROI” function in the Source Image>>ROIs context menu
- Added “[0-1]-Normalize Decay” option in Settings>>Fluorescence Decay>>Decay Pre-Processing and as a function in Decay Graph>>Process Plot context menu
- Added “All ROIs Average Lifetimes” to FLI Dataset menu

1.11.29 0.65 (2022-12-15)

- **Bug Fixes**

- Syntax error in Python Plugin doesn't crash Python session anymore (a LabVIEW bug)

- **New Features**

- Improved Error and Warning reporting for Python Plugins
- The demo Python Plugins coming with a new version are not reinstalled after the first run, allowing the user to remove them

1.11.30 0.64 (2022-12-04)

- **Bug Fixes**
 - Unknown error in Python is now followed by an automatic reset of the Python session
 - Fixed issues with passing String or Path parameter to Python Plugin
- **New Features**
 - Support for Python Plugin in Analysis>>Decay Graph and Analysis>>Source Image menus
 - Support for Python plugin user-input parameter description
 - New installation does not overwrite Python Plugin files if they are newer than the version provided with the installer (kept in the data folder as a zip archive)

1.11.31 0.63 (2022-11-26)

- **Bug Fixes**
 - Fixed bug in ROI Manager which showed a warning when the selected ROI ID > 10,000 (should only occur if ‘Show All ROIs’ is selected)
 - Check the Clipboard content before trying to paste it in the Plot Editor
- **New Features (Python Plugins API)**
 - A script’s target is defined in the script itself, not by the location of the script in a folder hierarchy (the latter is now ignored to give more freedom to users).
 - Implemented modified version of Python plugin interface using named tuples to pass core data in and out of Decay Graph plugins
 - Implemented additional returned dictionary to pass additional information back to AlliGator (including updates of AlliGator parameters)
 - Added “Python Plugins” button in Settings>>Plugins to open the Plugins folder
 - Changed installation directory to C:\Users\UserName\AppData\Local in order to not require Administrator rights to install AlliGator or to modify files in the Python Plugins folders
 - Added support for insertion of plugins in window menus and object menus, with one insertion point max per location
 - Added alligator.py type definition file
 - Changed Python plugin syntax to use ### for all lines intended for AlliGator decoding
 - Modified Output section to be similar to the Input section
- **Other New Features**
 - Removed Ctrl+V shortcut to paste the Clipboard in Plot Editor
 - Added 256 characters limit to ROI description output to Notebook (inactive in verbose mode)
 - Removed separate shortcut Ctrl+Shift+R for “Reload FLI Dataset Series” and used it instead to open the Image ROI Manager

1.11.32 0.62.2 (2022-11-08)

- **Minor Improvement**
 - Improved Background Subtraction File compatibility and better warning dialog

1.11.33 0.62.1 (2022-11-04)

- **New Features**
 - Reverted to Toeplitz Matrix inversion method for deconvolution
 - Added IRF Optimization parameters to Notebook output
 - Changed IRF Optimization range parameter to ns instead of fraction of range center

1.11.34 0.62 (2022-11-02)

- **Bug Fixes**
 - Fixed bug in Extrapolate Decay (failed when large vertical offset was present)
 - Fixed bug where Plot Extrapolation failed for a normalized decay
 - Fixed display glitch in Settings >> Gates to Keep
- **New Features:**
 - Added Export Plot and Data options in IRF Extraction parameters
 - Improved Optimal IRF extraction criteria
 - Changed display of Python plugin scripts and functions in menus by replacing all underscore characters by spaces
 - Changed the way a plot is passed to a Python plugin: instead of two 1D-arrays, a single 2 x N array is used
 - Added support for Python plugins acting on FLI Dataset (Analysis>>FLI Dataset menu)

1.11.35 0.61 (2022-10-23)

- **Bug Fixes**
 - Fixed a bug that added a carriage return in the name of extrapolated decay plots (which prevented saved plots to be properly loaded)
 - No Notebook message is output if plot rebinning cancelled
 - Fixed a few Notebook output formatting issues
 - Fixed error returned when a series of HDF5 dataset missing individual timestamps was selected
 - Fixed Files->Load menu bugs (menu items were not active)
 - Removed a number of error messages returned when the user cancels a file selection
 - Fixed a glitch in the dialog for Multi-ROI single Pixel analysis scripts
- **New Features**
 - Source Image Settings: Added option to remove hot pixels when loading SSx datasets

- Decay Graph: Edited menus (Decay -> Plot)
- Decay Graph: Better plot names and support for imported files without abscissa column
- Decay Graph: Added “Rebin Plot” menu item
- Decay Graph: Modified algorithm to extract IRF from decay with known lifetime, and added automatic smoothing and removal of outliers at the beginning of the IRF
- Fluorescence Decay Settings>>Advanced Analysis: Added new IRF extraction options
- Added average lifetime information for 2-Exp fit in Notebook output

1.11.36 0.60.4 (2022-08-18)

- **Bug Fixes**

- Fixed “Save All/Selected Phasor Plots” Phasor Graph menu items
- Fixed bug preventing Single Phasor Calibration Load/Save/Clear menu items to work
- Fixed bug preventing from loading Single Phasor Calibration or Phasor Calibration Series if the frequency was identical to the current one (should be the opposite. Worked fine for Calibration Map)
- Fixed issue with Series Phasor Analysis missing corresponding metadata

1.11.37 0.60.3 (2022-08-15)

- **Bug Fixes**

- Restored “Export Add'l Phasor Plots Data (ASCII)” menu of Phasor Graph

- **New Features**

- Grouped basic phasor plot save functions under “Save Plots (ASCII)” sub-menu
- Grouped phasor plot save functions under “Save Phasor Plots (.phplot)” sub-menu

1.11.38 0.60.2 (2022-08-13)

- **Bug Fixes**

- Fixed display glitches in Multi-ROIs Single-Pixel Analysis dialog window
- Speed up of Phasor Information Display in Phasor Plot when shift-hovering over the Source Image with the mouse

- **New Features**

- Replaced “Phasor Ratio Calculation” checkbox by “Phasor Ratio Type” radio buttons: Intensity-weighted or Amplitude-weighted

1.11.39 0.60.1 (2022-08-04)

- **Bug Fixes**

- Fixed bug that would not display the newly added file in the Batch Export Window

1.11.40 0.60 (2022-08-03)

- **Bug Fixes**

- Fixed bug in Image Python Plugin handling of output image
- Fixed error generated when loading single TIFF Images of different size in succession
- Fixed bug in Image Type selection (which would allow selecting a non-existing image)
- Removed the (bogus) ability to drop an ASCII plot in the Phasor Graph. Only phplot files (Phasor Plot Data) can be imported in the Phasor Graph
- Removed bogus Lifetime plot created when performing Data Series Phasor Analysis
- Fixed bug that would erase the last Phase Lifetime plot when performing a new Data Series Phasor analysis

- **New Features**

- Modified Mask Image to ROIs to allow handling binary images (as was the case up to now) as well as “labeled” mask images
- Removed “Export Phasor Graph Data” menu items (they were equivalent to “Save Phasor Plot” items, now renamed “Save Phasor Plot Data”) to disambiguate them from the standard ASCII plot menu items (“Save Plot(s)”).
- Added option to use the current phasor calibration/IRF(s) for Multi-ROI Single-Pixel Phasor or NLSF Analysis script

1.11.41 0.59 (2022-08-01)

- **Bug Fixes**

- Fixed bug generated when loading single TIFF Images of different size in succession
- Fixed Sticky Help Ballon in Batch Export Window

- **New Features**

- Added support for 16-bit Mask Image
- Modified Mask Image to ROIs to allow handling binary images (as up to now) as well as labeled mask images
- Added “Paste Plot(s) from Clipboard” in Decay Graph and other Graphs

1.11.42 0.58 (2022-07-28)

- **Bug Fixes**
 - Added missing “SYNC Period” in exported HDF5 files
- **New Features**
 - Added (optional) Help Balloon in Batch Export Window
 - Implementation of Multiframe loading and batch exporting (ptu files)
 - Added “Dataset Timestamp” field in AlliGator HDF5 File format v0.6.1
 - Added warning dialog when requesting to “Show All ROIs” when N > 10,000
 - Optimization of ROI to Pixel ROIs action

1.11.43 0.57 (2022-07-21)

- **Bug Fixes**
 - Fixed bug in Phasor Graph->Export Phasor Plot(s) Data, which would not output the full header line
 - Fixed initialization of “Gates to Use” parameter in Settings->Data Information
 - Fixed “jumping” scrollbar in Notebook
 - Fixed disappearing selection when using context menu in Notebook
- **New Features**
 - Code refactoring for loading/closing speed up
 - Added font size options in Notebook

1.11.44 0.56.4 (2022-07-17)

- **Bug Fixes**
 - Fixed display of calibrated L_N[W] SEPL
 - Fixed bug in loading “Gate Separation” for Becker & Hickl .std files
 - Fixed a bug in “Skip Gate” file loading options, which was swapping “start” and “end” values

1.11.45 0.56.3 (2022-06-30)

- **Bug Fixes**
 - Deprecated “Whole Image Phasor Scatter Plot” in Analysis->Phasor Graph, as it is now done by selecting a ROI encompassing the whole image and converting it to single-pixel ROIs, followed by Analysis->Dataset->Multiple ROIs Analysis->All ROIs Phasor Analysis->Non-Interactive (Fast) [Ctrl+Shift+A]

1.11.46 0.56.2 (2022-06-29)

- **Bug Fixes**
 - Fixed a bug preventing updating the SEPL harmonic frequency
- **New Features**
 - Added context menu item to SEPL:Harmonic Frequency to set it to the Dataset Phasor Frequency
 - Default IRF period is now 12.5 ns (used to be 0, resulting in no SEPL being plotted for anything but the UC option)

1.11.47 0.56.1 (2022-06-27)

- **Bug Fixes**
 - Fixed a bug which would not properly update the Reference Lifetime Indicators in the Phasor Graph panel
- **New Features**
 - Updated “# Gates” parameter description to explain that it is ignored in most cases

1.11.48 0.56 (2022-06-26)

- **Bug Fixes**
 - Fixed bug in Sum of Datasets
 - Fixed bug related to gate arithmetic in case of single gate datasets
 - Fixed bug in Gate Name storage after user selection dialog
 - Fixed bogus first Intensity Time Trace point in Phasor Series Analysis
 - Fixed unneeded ROI output to Notebook when Decay Graph & Phasor Graph are inactive
 - Fixed “sticky” Series Slide when loading an individual dataset
 - Fixed “busy” icon and UI sluggishness when adjusting Image Histogram cursors
 - Fixed Image Contrast Image Type
 - Fixed incorrect SEPL plot name in Phasor Graph
 - Fixed a bug in “ROI to single pixel ROIs” which would not remove the original ROI unless it was the first in the list
 - Fixed bug preventing loading FLI dataset files of different type during a single AlliGator session
 - Fixed released .ini file
 - Removes -Inf, Inf and NaN values from the Image Histogram
 - Removes -Inf & +Inf Sum of All Gates pixel values from ROI Decay calculation
 - Does not create empty decay plots (when no valid pixels)
 - Fixed UI update issues (busy icon, status message, file path controls, SS3 gate selection, etc.)
- **New Features**
 - Added Splash Screen on launch and shutdown

- ms precision for date/time stamps in Notebook messages
- Added dialog when loading single image (Mask or White Light)
- Reorganized AlliGator menu bar
- Removed “Average Dataset Sum” checkbox in Settings->Source Image (replaced by two menu items in Analysis->Dataset Series)
- Added option to save a “Phase Lifetime vs X” scatter plot during “Multi-ROI Single Pixel Phasor Analysis”, where X is in {None, Total Intensity, Background-subtracted Intensity, Amplitude, Background, Fitted Lifetime (Bkgd Subtraction)}
- AlliGator file loading refactoring to support dual-channel dataset
- Removed Import I/m/phi dataset from File menu
- Renamed Other Files->Open White Light Image and Open Mask Image to Other Files->8- or 16-bit White Light Image and 8-bit Mask Image
- Added Python plugin support for Source Image & Decay Graph
- Added Settings->Plugins panel
- Added “Export JSON String to Clipboard” right-click menu item to Settings controls (this string being used to pass each parameter to Python plugins).
- Added “Send” AlliGator Parameters JSON Strings to Clipboard button in the Settings->Plugins tab
- Implemented support for internal AlliGator parameter passing to Python Plugins in addition to user-specified parameters
- Added support for python doc string and sending them to the Notebook when using the ‘H’ key pressed while selecting a plugin menu item

1.11.49 0.55 (2022-05-13)

- **Bug Fixes**
 - Fixed Image Histogram failure when NaN pixel values were encountered
 - Fixed bug in “Define Shortest Lifetime as” Settings which would always set it to Reference 1
- **New Features**
 - Added $G2/INT \cdot \max(INT)$ gate arithmetic option

1.11.50 0.54 (2022-05-12)

- **New Features**
 - Added “Define Shortest Lifetime as” (Reference 1, Reference 2) radio button to Settings->Phasor Graph window to tell how to define which reference is 1 or 2 when using UC/Fit intersection or Minor/Major Axis/UC Intersection in either Phasor Plot or Phasor Graph

1.11.51 0.53.1 (2022-05-10)

- **Bug Fixes**

- Fixed erroneous saving of Phasor Plot image with overlay irrespective of the selected option
- Fixed erroneous Decay Name when using File Background correction (was set to the background file name)
- Fixed erroneous Dataset Information when using File Background correction (showed background file information: now shows both)

1.11.52 0.53 (2022-05-07)

- **Bug Fixes**

- Removed Graph Index Display because of conflict with plot identification

- **New Features**

- Implemented 1-Normalize, Shift and Fold decay in the Decay Graph context menu

1.11.53 0.52 (2022-05-05)

- **Bug Fixes**

- Fixed a bug in 2-Exp Fit which converted it into a 1-Exp fit
- Fixed a bug in the Drag & Drop behavior of the ROI Managers
- Fixed a bug in Settings>>Fluorescence Decay>>Decay Pre-Processing, which did not preserve either the Pile-up Correction flag or the Max Value.
- Corrected Example String in single variable and 2-variables formula input dialogs
- Fixed update bug in Plot Editor “Fill Array with Ramp” dialog
- Fixed a bug in Export Phasor (multiple) Plot Data to ASCII where the header might be incomplete
- Fixed longstanding bug in Phasor Plot computation when decay pre-processing is selected

- **New Features**

- Added Gate Arithmetic Enum to combine SS3 gates: None, INT-G2, ($\langle \text{INT} \rangle + \text{INT}$)/2-G2, $\text{G2} + (\langle \text{INT} \rangle - \text{INT})/2$
- Moved Background File Subtraction parameters from Decay pre-processing to Data Information
- Added Plot Legend Index Display to Graphs to complement scrollbar
- Fixed missing Notebook message when saving images with overlay
- Fixed ambiguity when saving SGL image to 16-bit TIFF

1.11.54 0.51 (2022-03-07)

- **Bug Fixes**
 - Fixed Error in reduced Chi2 expression (Chi2/DOF, was Chi2/N)
 - Fixed NLSF Fit
 - Fixed Notebook backup issues with new logic and added message bar at the bottom to inform on most recent backup
- **New Features**
 - Upgraded to LabVIEW 2021 SP1
 - Changed Fit Weight option from boolean to enum (unweighted, 1/Variance, Best of Both)
 - Uses normalized decay fit internally but returns scaled parameters
 - Now outputs correct reduced Chi2 in Notebook (was RMSE)
 - Outputs Guess Parameters even if fit fails
 - Added Gate Names ring control to Settings (for multigate FLI datasets such as encountered in SS3)
 - Added export of intensity and amplitude-phasor ratio and average lifetime in export tool
 - Implemented Batch Conversion to HDF5 and TIFF series

1.11.55 0.50 (2022-01-31)

- **New Features**
 - Added ProcessID to Notebook backup file name to distinguish between processes
 - Added “Live” Highlight of Source Image pixels in Phasor ROI (H key pressed while moving selected ROI in Phasor Plot)

1.11.56 0.49 (2022-01-30)

- **New Features**
 - Replaced Image Sum readout in nested loops by Array indexing (Image Sum array created upon creation)
 - Implemented # Logical Processors Used in Settings, in order to control CPU usage for resource-demanding functions such as NLSF
 - Compares unweighted & weighted fit results when weighted fit is requested, returning the best of both

1.11.57 0.48.4 (2022-01-14)

- **Bug Fixes**
 - Fixed bug in Clear Graph in Histogram Windows
- **New Features**
 - Added LED indicator when IRF options have been defined in Multi-ROI Single-Pixel Scripts
 - Added Set Options button (and LED) for dataset (for consistency)
 - Added message to Notebook after saving it (if it fails, a warning is printed)
 - Updates Parameter Range when changing selected parameter in Decay Fit Parameter Map

1.11.58 0.48.3 (2022-01-14)

- **Bug Fixes**
 - Fixed transition from Accumulated to Single File and back
 - Fixed bug that prevent intensity time trace from being computed when no decay pre-processing options were selected

1.11.59 0.48.2 (2022-01-11)

- **Bug Fixes**
 - Fixed ROI Statistics error when at least one of the pixel value is negative (now removed from SNR statistics)
 - Fixed ROI Color update at startup
 - Fixed bug in “Export ROI Data as ASCII” and “Export All ROIs Data as ASCII” in Decay Fit Parameter Map

1.11.60 0.48.1 (2022-01-08)

- **Bug Fixes**
 - Fixed a bug in Sum All Datasets in Series that prevented it from working
 - Notebook fails to save when path is longer than 259 characters

1.11.61 0.48 (2022-01-07)

- **Bug Fixes**
 - Fixed a bug which prevented the “Save As...” menu of the Notebook to work
 - Fixed a bug in the Notebook that made it non-persistent
- **New Features**
 - Switched to HDF5 format for IRFs & Fit Data Map
 - Lifetime Analysis panel renamed Lifetime & Other Parameters Analysis

- Image ROIs (stored in the ROI Manager) are now shown in the Decay Fit Parameter Map
- Support for exporting Selected ROI(s), All ROIs or All Map Data to Lifetime & Other Parameters Analysis Graph
- Replaced Phasor Calibration checkboxes replacement by Pull-down menu
- Added ROI ID selection and corresponding UI
- Implemented Multi-ROIs Single-Pixel NLSF Analysis Script & Implemented Multi-ROIs Single-Pixel Phasor Analysis Script
- Added IRFs & Fit Data Map export
- Implemented single ROI pixel-wise phasor calculation and phase lifetime export
- Added “Delete All Unselected ROIs” to ROI Manager and Image context menu
- Changing the selected ROI (in Source ROI Manager) updates the Local Decay Graph window if the ROI is a point
- IRF number does not need to be identical to decay number in All ROIs Decay Fit (No-interactive, fast, Individual IRF)
- Added “Fold Decay” preprocessing option (parameter: # Folds)
- Added Analysis>>Series Analysis>>Show Dataset Series Sum menu item
- Updated Settings parameters change triggering Phasor Plot update flag (background-correction related parameters)
- Added Histogram Options to Settings>>Miscellaneous: used for scripts
- Added Export All Maps Data as ASCII menu item to Decay Fit Parameter Map
- Added “Export ROI Data as ASCII” and “Export All ROIs Data as ASCII” menu items
- Implemented IRF Options in NLSF/Phasor Script
- Added Color Boxes to Select ROI Color for Source Image/Phasor Plot and Decay Fit Parameter Map
- Linked Decay Fit Parameter Map ROI to Source Image ROI

1.11.62 0.47 (2021-12-18)

- **New Features**

- Added mode, median and SNR statistics to ROI Statistics output to Notebook
- Added option to use a First Gate index larger than Last Gate index when using the “Gates to Keep” option. In that case, the gate images (F, F+1, ..., N-1) are lost

1.11.63 0.46 (2021-12-11)

- **New Features**

- Changed Phasor Graph menu item: Phase Lifetime Trajectory >> Phase Lifetime Series
- Added Save Phasor Plot Add'l Data submenu: Save Single Phasor Plot Add'l Data, Export Selected Phasor Plots Add'l Data, Export
- Added Export Single Phasor Plot Data, Export Selected/All Phasor Plots Data

1.11.64 0.45.2 (2021-12-03)

- **New Features**
 - Updated dialog window message when exporting average lifetime map from Phasor Plot

1.11.65 0.45.1 (2021-12-01)

- **New Features**
 - Tweaked amplitude fit parameters adjustment when selecting “Use Last Fitted Parameters”
 - Changed IRF normalization for convolution so that its integral is equal to 1 => fitted parameters (baseline, amplitudes) are accordingly rescaled
 - Removed a few innocuous error messages showing up in verbose error mode

1.11.66 0.45 (2021-11-30)

- **Bug Fixes**
 - Fixed update of Phasor Ratio References with Phasor Frequency (Phasor Plot & Phasor Graph)
 - Fixed a bug in Extrapolate Decay as part of Pre-Processing (Head & Tail fractions were reversed)
- **New Features**
 - Changed UI behavior of “Clear Source Image Overlay” and “Clear Phasor Overlay” buttons, by removing adjacent checkmarks and moving all options to right-click menu. See context help for description.
 - Added option to define the Phasor Ratio References by Phase Lifetime value (in Phasor Graph)
 - Added dashed line connecting references in Phasor Plot
 - Added support for decay pre-processing in the Local Decay Window
 - Changed style of warnings in Notebook
 - Implemented baseline and amplitude fit parameters adjustment when selecting “Use Last Fitted Parameters”

1.11.67 0.44.3 (2021-11-12)

- **Bug Fixes**
 - Fixed Calibration Map change detection, resulting in subsequent speed up of phasor plot update
 - Fixed bug preventing loading of Calibration, Calibration Series or Calibration Map
- **New Features**
 - Changed the Phasor Ratio/Amplitude output from f2 to f1 (fraction of reference 1). The other derived quantities (e.g. average lifetime) were computed correctly and therefore remain unchanged. This definition is now made clear in Notebook outputs or indicator captions.
 - Changed some default Fit Options Termination Criteria ($1E\pm 6 \rightarrow 1E\pm 9$) to improve convergence

1.11.68 0.44.2 (2021-11-11)

- **Bug Fixes**
 - Fixed fit parameters map file loading error
 - Fixed error when drag & dropping plot when the visible Graph does not support drag & dropping
 - Fixed bug in 2-Exp NLSF introduced in v0.42
- **New Features**
 - Minor menu edits
 - Added Fit Options & Parameters to IRFs & Fit Data Map output

1.11.69 0.44.1 (2021-11-07)

- **Bug Fixes**
 - Fixed missing Pixel-Normalize Decay step in Phasor Array Calculation
 - Fixed problem with computing Phasor Plot when a large number of ROIs is defined

1.11.70 0.44 (2021-11-06)

- **Bug Fixes**
 - Minor fixes to handling of Fit Parameter Constraints/Guess Parameters and Displayed Fit Parameters arrays in Settings>>Fluorescence Decay>>Fit Parameters
 - Fixed cases where no ROI exists
- **New Features**
 - Opens the Notebook on startup
 - Added Major/Minor Axis option in UC Intersection in Phasor Plot and Phasor Graph
 - Added Linear Fit - UC Intersection in Phasor Pot
 - Changed Phasor Plot Reference manipulation approach: now uses keyboard key 1 and 2 to select reference 1 and 2, the buttons being highlighted when used
 - Added “Selected Phasor Plot ROI(s)” and “All Phasor Plot ROIs” option to analyze the Phasor Plot (e.g. Reference calculation)
 - Added Selected Phasor Plot ROI(s)” and “All Phasor Plot ROIs” option for Linear Fit/UC intersections
 - Added support for Phasor Plot ROI file Drag & Drop

1.11.71 0.43 (2021-11-03)

- **New Features**

- Added “Selected Image ROI(s)” and “All Image ROIs” option to compute the Phasor Plot
- Simplified phasor plot computation
- Implemented faster smoothing algorithm (at the expense of less accurate pixel rejection)
- Added option for smoothing algorithm
- Added “Phasor Ratio Color-Coded Quantity” parameter in Phasor Plot Settings, to select between Phasor Ratio and Average Lifetime when overlaying the phasor plot results onto the Source Image
- Export of Phasor Ratio Map and Average Lifetime Map added to Phasor Plot contextual menu

1.11.72 0.42.3 (2021-10-29)

- **Bug Fixes**

- Fixed bug in Gates to skip/Gates to keep calculation

1.11.73 0.42.2 (2021-10-27)

- **Bug Fixes**

- Corrected bug in amplitude weighted average phase lifetime calculation (Phasor Graph)
- Fixed non-functioning “Save Phasor Plot Image with Overlay” menu item (Phasor Plot)

1.11.74 0.42.1 (2021-10-21)

- **Bug Fixes**

- .ptu loading files touch-up (inlining, code separated from VI, shift register initialization) to improve performance
- Fixed default filename when exporting HDF5 dataset

1.11.75 0.42 (2021-10-158)

- **Bug Fixes**

- Fixed a bug in Fit Function weight calculations

- **New Features**

- Added “Define Gates to Keep” option (in addition to Gates to Skip)
- Removed Error in NL Fit if the Hessian cannot be inverted
- Fit Options are now used throughout AlliGator (e.g. Histogram)

1.11.76 0.41 (2021-10-10)

- **Bug Fixes**
 - Fixed Drag & Drop of folder of folder of gate images
- **New Features**
 - Removed White Light Image Histogram (fused with Fluorescence Image Histogram)
 - Support for Drag & Drop of Mask, White Light Image
 - Support for Drag & Drop of individual Phasor Plot (phplot) in Phasor Graph
 - Support for Drag & Drop of calibration, calibration series & calibration map
 - Support for Drag & Drop of IRFs & Fit Data files

1.11.77 0.40.1 (2021-10-08)

- **Bug Fixes**
 - Fixed a bug in Compute Average Phasor (Selected Plots) that would remove the first selected plot if the UC was not selected
- **New Features**
 - Modified zero-weight replacement to use interpolated values whenever possible (instead of 1)
 - Added Average Lifetime to Phasor Plot & Phasor Graph display and Notebook output when references are defined

1.11.78 0.40 (2021-10-07)

- **New Features**
 - Added transparent option for Phasor Plot References and Boundary
 - Added Phase Lifetime SDV output for Average Phasor (as well as Modulus and SDV)
 - Added Compute Average Phasor (Selected Plots) to Phasor Graph
 - Added Phasor Ratio (or Amplitude Ratio if selected) as Notebook output for the previous operations

1.11.79 0.39.4 (2021-10-06)

- **Bug Fixes**
 - Fixed context menu for phasor frequency and initialization in SEPL Settings
- **New Features**
 - Update Phasor Frequency at startup
 - Added Reference Point Radius parameter for Phasor Plot

1.11.80 0.39.3 (2021-09-30)

- **Bug Fixes**
 - Corrected a bug in Multiple ROI Analysis that could result in some ROIs' data to not be processed

1.11.81 0.39.2 (2021-09-29)

- **Bug Fixes**
 - Corrected bug in Export Decay to Phasor Graph menu item added to Decay Graph

1.11.82 0.39.1 (2021-09-28)

- **New Features**
 - Added Create ROI(s) from Pixel with Intensity over Min

1.11.83 0.39 (2021-09-27)

- **New Features**
 - Modified description of fit parameter map parameters
 - Added Export Bounding Cursors Definition to Notebook menu item to Decay Graph
 - Bounding cursors follow multiple/all plots
 - Added separate Pile-up Correction option for Background File
 - Does not show the Create Phasor Plot Dialog if Phasor Graph is inactive
 - Added Export Decay to Phasor Graph menu item added to Decay Graph
 - Moved some items in Decay Graph menu
 - Output Decay Fit results even in case of convergence or other failure (with error message)
 - Added verbose message for ROI manipulation
 - Added more colors to Notebook
 - Force file dialog when saving IRFs & Fit Data

1.11.84 0.38.1 (2021-09-14)

- **Bug Fixes**
 - Fixed bug in standard File menu based data loading

1.11.85 0.38 (2021-09-13)

- **Bug Fixes**

- Fixed a but preventing from aborting Playback
- Fixed a bug in loading ptu files
- Fixed a bug in Playback Script where the dataset index was not incremented
- Fixed Natural Frequency calculation to take into account the “User Decay Pre-Processing” flag
- Fixed progress bar display for parallel loops
- Corrected bug in Notebook output of Phasor Plot calculation (was returning number of valid phasors instead of invalid phasors)
- Corrected a bug in Phasor Plot calculation (phasor calibration map case) and simplified code
- Fix bug where a background constant subtraction was not applied when computing the Phasor Plot
- Corrected a bug in Phasor Amplitude Ratio calculation
- Cancel in Phasor Ratio abscissa input is now handled properly
- Fixed OpenG bug resulting in erroneous interpretation of .ptu files

- **New Features**

- Implemented drag & drop file/folder to open a file or a series
- Added Drag & Drop of ROI definitions
- Added support for Drag & Drop of plots
- Added option to create new Phasor Ratio plot in input dialog window
- Output of phasor frequency in multipoint Phasor Plot calculation
- Improved display precision for several numeric indicators
- Save Image or Phasor with Overlay now does not require computing any overlay
- Made error message when no Reference Decay/IRF is found clearer (it could be due to the “Use Local IRF” checkbox being checked)
- Changed menu item from Compute Phasor Plot’s Phasor Ratio to Compute Phasor Plot’s Average Phasor Ratio
- Added Phasor Plot name(s) used to compute references in Phasor Graph
- Improved Decay Extrapolation by increasjing weight of head part in case of periodicity
- Added overwrite warning in Notebook Save As...
- Improved extrapolated decay (fixed NL Fit Options, head weights, shotnoise)
- Phasor Explorer-related updates (using identical Settings Parameters for SEPL definition)
- Added “Store Cursor-defined Head & Tail Fractions” in Decay Graph context menu (to use in automated analyses)
- Restored Algebraic IRF deconvolution
- Added verbose Notebook message for Phasor Ratio reference updates
- Added “Phasor Graph Active” checkbox

1.11.86 0.37 (2021-08-16)

- **Bug Fixes**
 - Corrected bug in Use Single Plot Fitted Line/UC Intersections
 - Corrected bug in Multiple Plots Fitted Line/UC intersections which include UC in the calculation
 - Update Phasor Graph line if reference is updated
 - Fixed bug in Load Plot(s) which was missing partially empty columns
 - Modified Linear Fit in Phasor Ratio Graph to encompass min and max values
- **New Features**
 - Added Use Multiple Plots Fitted Line/UC Intersections to Phasor Graph contextual menu
 - Added Single Phasor Plot Phasor Ratio calculation

1.11.87 0.36 (2021-08-05)

- **Bug Fixes**
 - Does not reject Phasor Plot calculation if an individual phasor cannot be computed
 - Fixed Image contrast update frequency when moving mouse in the image histogram(s) by requiring the mouse to be down for this to happen
 - Fixed Preprocess Decay checkbox impact on calculations throughout AlliGator (was partially implemented)
- **New Features**
 - Added ability to view decay profile at location in Image Source
 - Added requirement to press the Left or Right Shift key for Phasor update at current location to actually update phasor information in the phasor plot panel (in order to avoid queueing too many events), when moving the mouse in either the Source Image or the Phasor Plot image

1.11.88 0.35 (2021-07-26)

- **Bug Fixes**
 - Fixed Clear ROI bug in Decay Fit Parameter Map Image
 - Fixed Square Gate Fit corner case issue
- **New Features**
 - Added Save/Load IRFs & Fit Data Maps
 - Added IRF to Local Decay Graph Window
 - Added Mouse Click requirement to update Local Decay Profile window
 - Added controls for Square Gate Fit Parameter Map
 - Improved Logistic Gate Fits
 - Implemented option to use local IRF for decay fits (manual)

1.11.89 0.34.4 (2021-07-23)

- **Bug Fixes**
 - Fixed progress bar problem
 - Fixed Settings>>Data Information export to Notebook

1.11.90 0.34.3 (2021-07-22)

- **Bug Fixes**
 - Fixed 2nd-pass fit approach in Decay Fit Parameter Map

1.11.91 0.34.2 (2021-07-22)

- **Bug Fixes**
 - corrected a bug in the Decay Fit Parameter Map output for 2-Exp decays
- **New Features**
 - Implemented more robust clip decay for fit
 - Added NaN check in decay before fit
 - updated definition of derived fit parameters and added better description of their definition

1.11.92 0.34.1 (2021-07-01)

- **Bug Fixes**
 - Fixed bug where the periodicity option (and period) were not taken into account in “Extrapolate Decay” when extracting a decay
 - Fixed glitches in Settings>>Fluorescence Decay>>Decay Pre-Processings update of “Extrapolate Decay” options

1.11.93 0.34 (2021-06-28)

- **New Features**
 - Image Profile (Windows>>Image Profile) shows the selected parameter in the Decay Fit Parameter Map. It is plotted associated with the lifetime axis (to the right). You need to draw a line on the image for this to show anything and there are some “features” as far as the update goes, but it essentially does the job.
 - Local Decay Graph (Windows>>Local Decay Graph) shows the decay, fit and residuals at the point in the Decay Fit Parameter Map when the “Point” tool is selected (bottom of the list on the left of the map). Don’t forget to adjust the Intensity Axis to show the correct range of values.

1.11.94 0.33.7 (2021-06-27)

- **Bug Fixes**
 - Fixed a cosmetic bug with min/max decay parameters
- **New Features**
 - Added condition to retry fit in decay fit map: $R^2 < 0.95$

1.11.95 0.33.6 (2021-06-27)

- **Bug Fixes**
 - Progress bar edits
 - Minor cosmetic changes to Decay Fit Parameter Map behavior

1.11.96 0.33.5 (2021-06-26)

- **Bug Fixes**
 - Fixed bug in decay fit parameter output
 - Fixed missing Guess Parameters output to Notebook in Settings>>Export Settings to Notebook
 - Fixed Fit Parameter Map color scale update
 - Fixed missing Settings string outputs
- **New Features**
 - Added Use Decay Pre-Processing Option
 - Turned back parallel computation on for decay fits
 - Added 2nd round of fits for decay fit parameter map, using successful neighboring fits

1.11.97 0.33.3 (2021-06-24)

- **Bug Fixes**
 - Fixed some newly introduced Settings memory bug
 - Fixed HDF5 Dataset Parameter saving

1.11.98 0.33.2 (2021-06-24)

- **Bug Fixes**
 - Added dlls needed for h5labview in build

1.11.99 0.33 (2021-06-22)

- **Bug Fixes**

- Fixed an inactive menu item in Settings
- Fixed error when ROI pixel is outside image
- Fixed a bug in Fit Decay parameter passing for fitted decay calculation
- Fixed bug where 1-Normalize Decay was ignored
- Corrected bug in Save Plot dialog
- Fixed bug in reading HDF5 files version 0.4 (introduced with file version 0.5)
- Fixed Pile-up Correction update bug in Settings
- Corrected a bug in Accumulate/Average Datasets where the number of gates was used instead of the number of datasets
- Corrected UI bug which would remember Background Dataset as last used files
- Fixed bugs in Histogram Window
- Corrected typo in Define ROI(s) above Min Peak menu item
- Fixed bugs in Phasor Image calculation
- Fixed high CPU usage in Source Image local Phasor display
- Fixed missing menu items in Phasor Graph
- Fixed missing Operation Order parameter in Settings
- Fixed Phasor Frequency synchronization (removed Phasor Calibration Frequency in Settings)

- **New Features**

- Modified IRF & Decay Parameter name in SEPL tab: “IRF Period”
- Added IRF Period to Fluorescence Decay >> Basic Analysis and Fluorescence Decay >> Fitting
- Right-click menu allows setting IRF Period in Fitting and SEPL to the same value as in Basic Analysis
- Right-click menu allows Setting IRF Period in Basic Analysis to Laser Period
- Implemented PSED + IRF convolution fit
- Implemented 2-Exp periodic convolution fit
- Added “Use Last Fitted Parameters” to Guess Parameters Tab in Fluorescence Decay Tab & associated logic
- Added plot clipping information in Decay Fit output
- Added Guess Fit Parameters in Decay Fit output
- Added repetition of fit to improve convergence
- Added All ROIs Decay Fit scripts (interactive and fast)
- Added optional export of tabulated fit results in non-interactive mode
- Renamed Settings>>Decay Analysis>>Fitting to Fit Options, and moved some controls to >>Fit Parameters
- Added Max Fit Iterations and Tolerance as part of the Fit Options
- Added Oval and Point ROI Grid definition

- Added all Termination Conditions to Fit Options
- Added Support for version 0.5 of HDF5 File Format
- Added automatic switch to unweighted fit when performing offset fit, followed by weighted fit pass if this is the user choice
- Added check to avoid mistakenly overwriting plot files.
- Modified Decay Extrapolation to support periodic decays
- Added “Use Periodicity” checkbox in Settings>>Fluorescence Decay>>Basic Analysis under “Extrapolate Decay”
- Added “Period” parameter to Extrapolate Decay (in case of periodicity)
- Using tail and head part of the decay for periodic decay extrapolation
- Added cursors option to define head and tail fraction for extrapolation (only interactively)
- Added Head Fraction as a Settings parameter
- Outputs fit range when using min/max percentile
- Suggested HDF5 FLI Dataset File Name now returns current dataset and folder
- Added output of number of averaged/accumulated datasets in Notebook
- Updated Save FLI Dataset as HDF5 to follow the new HDF5 File format (and to implement compression more efficiently)
- Added All IRF Analysis (fast & slow) scripts
- Implemented All ROIs Decay Fit Non-Interactive (Fast + Individual IRF)
- Added Progress Bar to main window (supported in All ROIs Decay Fit Non-Interactive (Fast + Individual IRF))
- Added Status Indicators for Reference/IRF and Individual IRFs Defined
- Added Decay Fit Parameter Map
- Implemented contextual menu for Decay Fit Parameter Map
- Implemented Use Image Brightness in Decay Fit Parameter Map
- Added Merge All ROIs in Source & Phasor Plot Image ROI Manager
- Added Invert Binary Mask Option in Settings>>Source Image
- Added busy icon display for potentially lengthy tasks
- Implemented Export Phasor Ratio Map to ASCII in Source Image
- Added support for binning of white light image and ROI mask
- Added Send Map Data to Lifetime Graph shortcut menu
- Added support for binning of white light image and ROI mask
- Added Send Map Data to Lifetime Graph shortcut menu

1.11.100 0.32 (2021-04-01)

- **Bug Fixes**

- Handling of incomplete image gate series in HDF5
- Corrected a bug in partial loading of HDF5 Datasets
- Corrected a bug in the computation of natural frequency in Settings
- Fixed popping-up dialog when some mundane error was encountered
- Fixed default settings bug in Histogram Fit
- Fixed bug in v0.3.1 support
- Fixed a LV bug with IMAQ Read Image.vi which misreads U16 TIFF images
- Fixed a bug with Time Gate Slider
- Updated dependencies after Phasor Explorer project update

- **New Features**

- Added support for FLI HDF5 file version 0.3.1 adding Image Information (cropping)
- Added support for FLI HDF5 v0.4
- Moved Calibrations settings to a separate panel in Settings
- Added SEPL in Phasor Graph & Phasor Plot and SEPL Parameters panel in Settings
- Removed “Show Ticks” for SEPL (common option with UC)

1.11.101 0.31 (2021-01-29)

- **Bug Fixes**

- Corrected a few bugs/cosmetic issues in HDF5 file loading
- Corrected a bug in Time Series loading that prevented proper extraction of file name root in some special cases
- Fixed “Phasor Plot Update Needed” flag erased by Settings Window

- **New Features**

- Support for HDF5 FLI dataset v 0.3
- Handling of incomplete image gate series in HDF5 file
- New experimental IRF deconvolution (undocumented in this version)

1.11.102 0.30 (2020-08-12)

- **New Features**

- Implemented compression of HDF5 FLI dataset

1.11.103 0.29 (2020-08-11)

- **New Features**

- Added Phasor Ratio Display Range slider to afford better control of the phasor ratio color map
- Implemented Use UC/Axis of Inertia Intersections as References in the Phasor Plot
- Added support for PicoQuant .bin files

1.11.104 0.28 (2020-08-03)

- **Bug Fixes**

- Support for rootless file name series (eg. 1.hdf5, 2.hdf5, etc.)
- Phase Lifetime vs Intensity Scatterplot now rejecting pixels with NaN phase lifetime values
- Minor Bug Fixes
- Some code refactoring

- **New Features**

- Introduction of an Image Profile tool to visualize Intensity/Phase Lifetime and Phasor Ratio
- Introduction of Phasor Harmonic Manager to handle harmonic data (currently only supporting 2-component Weber analysis)
- Output of Calibration information to the Notebook (including X & Y Resolution) when computing a Phasor Plot
- Display of Phasor Ratio/Amplitude at the mouse location in the Source Image/Phasor Plot
- Update of Phasor Ratio References when the phasor harmonic is changed
- Minor cosmetic changes

1.11.105 0.27 (2020-07-06)

- **Bug Fixes**

- Fixed bug preventing Histogram calculation in Graphs
- Fixed a bug where the Phasor Plot analysis functions did not take into account the conditions used to compute the Phasor Plot (e.g. min intensity)
- Fixed a bug of invisible Lifetime Graph tipstrip introduced in a recent release
- Minor Bug Fixes

- **New Features**

- Added Phasor Plot ROI resizing with Shift-Click
- Added Phasor Ratio color map display in Settings when interpolation is used
- Added constant baseline subtraction method
- Added Pixel-Normalize Decay option
- Modified Histogram Fit to use weighted fits and better estimates of the SDV of multi-Gaussians models
- Minor cosmetic changes

1.11.106 0.26 (2020-06-28)

- **Bug Fixes**
 - Fixed bug preventing pixel intensity conditions from being used for Phasor Plot calculation
 - Cosmetic and minor Bug Fixes
- **New Features**
 - Improves .set and RecSettings.txt decoding (LaVision dataset)
 - Added display of Phasor at Source Image pixel (in the Phasor Plot tab)
 - Support of stored ROI resizing/moving for the Phasor Plot (shift-click)
 - Added Phase Lifetime/Phasor Ratio Histogram and Phase Lifetime vs Pixel Intensity calculations in the Phasor Plot
 - Settings>>Phasor Plot now shows the color scale used when interpolation between the two references colors is used.
 - Minor cosmetic changes, including new layout of main menus

1.11.107 0.25 (2020-06-22)

- **Bug Fixes**
 - Fixed non-functioning Playback and Loop scripts
 - Corrected a bug preventing decoding of LaVision .set metadata files
 - Minor Bug Fixes
- **New Features**
 - Support for PicoQuant .ptu FLI datasets
 - Added a # Gates parameter in Settings>>Fluorescence Decays to define the number of bins to use for .ptu files
 - Added Right-Click menu for the Phasor Frequency controls to allow setting it to $1/T$ or n/T , where T is the laser period
 - Added Save Image Data as TIFF right-click option for Source and Phasor Images
 - Added Save Dataset as TIFF Series File menu item
 - Minor cosmetic (notably File menu) changes

1.11.108 0.24 (2020-06-12)

- **Bug Fixes**
 - Fixed a bug preventing proper loading of Gate Image Folder
 - Fixed sticky Phasor Plot Reference in Phasor Plot
- **New Features**
 - Added Phasor Plot ROI Manager Window mirroring the functionality of the Source Image ROI Manager Window

- Added output of Phasor References to Notebook when choosing “Use Segment Extremities as References”
- Added “Use UC/Segment Intersections as References” to Phasor Plot menu
- Added “Show UC Ticks” in Phasor Plot and Phasor Graph Settings
- Added UC tau ticks & labels to Phasor Plot and Phasor Graph
- Minor cosmetic changes

1.11.109 0.23.1 (2020-05-19)

- **New Features**

- Added support for compressed B&H sdt files
- Better formatting of complex data output in the Notebook

1.11.110 0.22 (2020-05-17)

- **Bug Fixes**

- Diverse inconsistencies in LED indicators have been fixed
- Fixed Reset Settings to Default in Settings
- Fixed Phasor Color Map Picker Save/Load menu

- **New Features**

- Image Binning can now be used to bin a dataset when loading it
- Phasor Frequency control added to the Phasor Plot tabs of AlliGator and Settings

1.11.111 0.21.1 (2020-05-11)

- **Bug Fixes**

- Fixed a bug preventing from exporting the Complex Phasor Data to file

1.11.112 0.21 (2020-05-10)

- **Bug Fixes**

- Avoid discontinuities in square-gated IRF fits
- Corrected a bug resulting in gate image order scrambling and/or image mask creation failure
- Plot Editor improvement
- File: Fixed Open Folder not remembering the last opened folder

- **New Features**

- Paste ROI Description from Notebook (or elsewhere) to create a ROI
- Decay Graph: IEEE square pulse analysis
- Script: Square Pulse Characteristics Map

- Import: SS1 Loading modified to account for older formats
- Phasor Plot: now includes Full Decay Preprocessing
- Phasor Plot: Parallelization to speed up processing
- Phasor Plot: Added Phasor Ratio Reference mouse control
- Phasor Plot: Added (m, phi), tau_phi and tau_m display
- File/Phasor Plot: Export Complex Phasor Data menu item to export complex phasor matrix to csv file
- Phasor Graph: extended right-click menu features for phasor ratio reference manipulation
- File: Added support for uncompressed B&H .sdt FLIM datasets (and dataset series)
- **Cosmetic items**
 - Changed calibration & reference status indicator
 - Removed Reload Dataset button to prevent confusion
 - Phasor Map Color Picker cosmetic fixes and improvements

1.11.113 0.20.4 (2019-11-26)

- **Bug Fixes**
 - Corrected a bug in File Background Correction (leading to failure to correct background Max Image)
- **New Features:**
 - Added Simple & Complex Plot Formula in Decay Graph
 - Added Background Scaling Factor parameter to adjust amount of background file to subtract

1.11.114 0.20.3 (2019-11-21)

- **Bug Fixes:**
 - Fixed a second bug where the decay pre-processing order wasn't saved properly
 - Fixed a bug where the total intensity of a decay was not saved if no background subtraction operation was used
 - Corrected a bug in the Phasor Plot calculation in the absence of background correction
 - Skip Square Gated Background Correction if one of the parameters is invalid (extrema positions, gate duration, laser period)
 - Fixed new version download/installation
- **New Features:**
 - Added Settings menu/modified menu layout
 - Implemented Tilted Square Gated IRF Fit (and corresponding Map script)
 - Changed Open Image Folder dialog behavior to show folder content and remember last open folder
 - Improved error handling when loading datasets
 - Added output to the Notebook of # Gates loaded

- Improved ROI Grid creation speed
- Additional information added to Dataset Information Window
- SS1 File importing now out of beta
- Made SS1 Laser Frequency dialog skippable

1.11.115 0.20.2 (2019-11-04)

- **Bug Fixes:**

- Fixed a bug where the decay pre-processing order wasn't saved properly
- Fixed a bug where the total intensity of a decay was not saved if no background subtraction operation was used

1.11.116 0.20.1 (2019-10-23)

- **Bug Fixes:**

- Fixed output format of Square-Gated IRF Characteristics Map Script
- Fixed AlliGator not closing upon quitting

1.11.117 0.20 (2019-10-05)

- **Bug Fixes:**

- Corrected the definition of the “Fixed High Background Threshold” (Settings>>Source Image) to refer to the Intensity Histogram mode instead of the maximum intensity value
- Corrected formula for square-gated exponential decay amplitude in Background Correction

- **New Features:**

- Added “Phase Lifetime vs Fitted Lifetime”, “Phase Lifetime vs Amplitude” menu items in Phasor Graph
- Added Square-Gated IRF Fit model in Fluorescence Decay Graph
- Added Plot Histogram to Phasor Ratio Graph (to study phasor ratio distributions when analyzing ROI series)
- Added “Skip this Dialog” in the Create Phasor Plot Dialog Window
- Added “Reset” button in Settings>>Miscellaneous to restore skipped dialog windows
- Implemented Phasor Calibration Map in Phasor Plot
- Added Color Scale indicator in Settings>>Phasor Plot to display the user-selected custom Phasor Ratio color scale
- Minor cosmetic changes

1.11.118 0.19 (2019-09-15)

- **Bug Fixes:**
 - If a HDF5 Dataset has already been pile-up corrected, it will not be corrected again even if the option has been checked in the Settings.
- **New Features:**
 - Added user-defined decay pre-processing operations order
 - Background subtraction added to the pre-processing operations list (allows performing it after other operations)
 - Unified SS2 HDF5 and AlliGator HDF5 dataset loading (no need to “import” SS2 datafiles)

1.11.119 0.18.2 (2019-09-13)

- **Bug Fixes**
 - Corrected a bug which prevented plotting phasor graph series when one of the ROIs was invalid
 - Handled cases of duplicated ROIs when computing a phasor calibration map: duplicates are now removed before computing the map and do not result in an error.
- **New Features**
 - Added “Rotate” option flag for decay shift
 - Added default shift decay option controlled by “shift” parameter (no longer equivalent to no shift)
 - Added Phasor Ratio Color Map option
 - Added option to remove out-of-bound values in the Plot Editor (to clean up data points before histogramming)
 - Decay pre-processing order changed from: Revert/Normalize/Smooth/Shift-Rotate/Extrapolate/Straighten to: Revert/Smooth/Straighten/Shift-Rotate/Extrapolate/Normalize

1.11.120 0.18.1 (2019-08-05)

- **Bug Fixes:** invalid phasor calculation when no background correction was applied

1.11.121 0.18 (2019-06-19)

- **Bug Fixes:**
 - Fixed Load Phasor Plot Data bug preventing from loading older file versions
 - Check for phasor plot adequation to be used as phasor calibration map
 - Does not automatically apply calibration anymore to the last plot when calibration (or type) is changed interactively
 - Fixed various Phasor Plot Overlay and Phasor Calibration issues

1.11.122 0.18 beta* (2019-06-03)

- **Bug Fixes**

- Corrected bug preventing cumulative phasor plot computation
- Fixed a bug resulting in scrambled gate images after a few files with different number of gates are loaded

- **New Features**

- Handling of decays with min preceding max (for square gate background correction)
- Messages added for decay processing
- Added partial Settings export to Notebook
- Added Background-subtracted Intensity to decay metadata (this is what is now returned instead of the amplitude, when using Phase Lifetime vs Bkgd-subtracted Intensity plot)
- Added automatic style choice for histograms and fits
- Added option to select 8-bit frames when importing SS2 dataset
- Invert Plots Selection added to Lifetime Graph
- Added right-click menus to Update Phasor Plot & Erase Overlay button to allow selecting which items are updated (and avoid unnecessary recomputation of the phasor plot)
- Updated implementation of Image Brightness use for Phasor Overlay: the histogram cursor positions are now used, not the image min and max
- Implemented it for Phasor Ratio and Phasor Color Map
- Optimized Binned Phasor Plot calculation
- Added Phasor Calibration Map Definition/Loading/Saving in Phasor Graph
- Added Update/Clear Selected Plots Calibrations
- Removed “Save Phasor Plot as...” and made it the default function of “Save Phasor Plot” on Phasor Graph (dialog always shows up)
- Implemented Mouse Wheel and Alt-Rectangle Zoom on Source & Phasor Images
- Implemented Phasor Calibration Maps (# Gates Series) Script
- Implemented Calibrated Phasor Map Series Script

1.11.123 0.17.5 (2019-05-03)

- **Bug Fixes:**

- Handling of corner cases where a gate image contains a NaN
- Corrected message output when saving single plot
- Fixed exported color bar size glitch
- Fixed Notebook close and open (keeps track of saved status and file path)
- Various fixes in child windows (Histogram & Plot Editor)

- **New Features:**

- Minor changes to SS2 file import

- Added “Show/Hide Tool Panel” Graph shortcut menu item
- Added Phase lifetime vs Background Plot
- Added Transpose Plot in Lifetime Graph
- Added Delete/Insert Element in Plot Editor

1.11.124 0.17.4 (2019-04-04)

- **Bug Fixes:**
 - Minor fixes
- **New Features:**
 - Style changed to LabVIEW “Next Generation”
 - Added Low and High Percentile rejection criteria in Image Source Settings
 - Implemented faster (non-interactive) All ROI Analysis

1.11.125 0.17.3 (2019-03-22)

- **Bug Fixes:**
 - Better graph legend clean-up
- **New Features:**
 - “Straighten Decay” now accounts for background
 - Added “Straighten Decay” as a decay processing option in Settings

1.11.126 0.17.2 (2019-03-12)

- **Bug Fixes:**
 - Fixed Histogram right-click menu
 - Fixed delayed histogram bin input

1.11.127 0.17.1 (2019-03-12)

- **New Features:**
 - Added a Hide/Show Grid menu item for Graphs
- **Bug Fixes:**
 - Restored missing right-click menus in a couple of Graphs

1.11.128 0.17 (2019-03-10)

- **New Features:**
 - Plot Histogram function added to Intensity Time Trace and Lifetime Graph
 - Added support for HDF5 Dataset Series
 - Support for Playback of all types of Dataset Series
 - Graph Style change
 - Changed file format for phasor plots and ROIs to JSON (from XML) for speed up and size reduction
 - Added support for background HDF5 Dataset subtraction
- **Bug Fixes:**
 - Restored compatibility with v1 of Raw Phasor Plot Data file (XML)
 - Added ROI validity check before analysis
 - Corrected a bug resulting in a dialog when reloading HDF5 dataset
 - Corrected a bug preventing reloading of image folders
 - Fixed memory leak problem most noticeable when dealing with dataset series

1.11.129 0.16.2 (2019-03-01)

- Fixed a bug preventing reloading a folder of images
- Added busy animated icon for Phasor Plot calculation and ROI Highlighting

1.11.130 0.16.1 (2019-02-28)

- Added support for SwissSPAD 2 Dataset Series
- Added Verbose Error Mode option in Settings>>Miscellaneous for support and debugging purposes
- Converted the “Recycle” button of the Image Source to “Reload Dataset”
- Changed the icon of the “Clear Overlay” buttons on both Source Image and Phasor Plot (the latter operation now skips Phasor Plot recalculation, which is done using the other “Refresh Phasor Plot” button)
- Added Sum All Datasets in Series function (with an “Average Dataset Sum” option in Settings>>Source Image)
- Open/Save HDF5 Dataset functions added
- Minor Bug Fixes and cosmetic changes

1.11.131 0.15.2 (2019-02-19)

- Corrected a bug introduced in the Phasor Plot
- Fixed a bug to render Notebook persistent

1.11.132 0.15.1 (2019-02-17)

- Parallelism in Phasor Plot calculation
- Added “Reload” menu item for all file formats (to apply gate specification changes)
- Renamed “Decay Points to Skip” to “Gates to Skip”
- Renamed “Revert Gates” to “Reverse Gates”
- Added File Path to AlliGator Window Title for all files
- Minor bug fixes and cosmetic changes

1.11.133 0.15 (2019-02-09)

- Minor Bug Fixes and cosmetic changes
- Added option to represent phase lifetime vs total intensity or background-subtracted intensity (Phasor Graph)
- Settings: Moved “Laser Period” parameter to Fluorescence Decay tab and “Pseudo Phasor Add'l Parameters” to Phasor Graph tab
- Removed Ctrl+C and Ctrl+Shift+C shortcuts to avoid conflict with the standard usage
- Added Amplitude/Background/Lifetime parameter display in the Fluorescence Decay panel (from background subtraction, when relevant)
- Added “Square Gated Single-Exponential” background subtraction method
- Renamed “Compute Phase Lifetime Plot” into “Phase Lifetime Trajectory”
- Added “Phase Lifetime vs Intensity Scatter Plot” menu item to Phasor Graph menu
- Added Timing Information to Notebook for image folder
- Added Active Graph/Plot flag to Settings
- Added support for SS2 HDF5 File
- Added option to save/load AlliGator Settings to file
- Added Laser Period & Decay Offset to Settings>>Fluorescence Decay>>Basic Settings
- Option to plot Pseudo UC in the Phasor Graph and Phasor Plot (beta feature)
- Whole image phasor scatter plot activated

1.11.134 0.13 (2018-09-18)

- Added Image subtraction as one of the background correction modes
- Added Whole Image Phasor Scatter Plot Calculation
- Added optional guess fit parameters for decay fitting
- Added “Integration” mode for decay fitting of data obtained with large gates, in addition to the previous “Convolution” mode
- Minor Bug Fixes and cosmetic changes

1.11.135 0.12 (2018-03-29)

- Added pile-up correction option for Swiss SPAD data
- Added ROI Grid tool
- Added ROI to pixel ROIs tool

1.11.136 0.10.11 (2018-03-13)

- Corrected a bug in Fluorescence Decay Statistics
- Improved performance of Fluorescence Decay Statistics

1.11.137 0.10.10 (2018-03-11)

- Color scale min and max color (Source Image and Phasor Plot Image) are updated upon clicking the color scale
- PDF version of the manual (on the AlliGator Manual page)
- Added “Revert Gates” option to handle SwissSPAD 2 data (beta)
- Periodic Boundaries option (beta)
- Beta version option in About window
- Swiss SPAD data support (beta)
- Minor bug and cosmetic fixes

1.11.138 0.10.9 (2018-02-21)

- Color scale min and max color (Source Image and Phasor Plot Image) are updated upon clicking the color scale
- Added “Revert Gates” option to handle Swiss SPAD 2 data (beta)
- Added “Max Lifetime” parameter to be used in determining how far to compute the model decay (beta)
- Implementation of periodic boundary convolution 1-Exp fit (beta)
- Swiss SPAD 1 support (beta)
- Added beta version option in About... window
- minor bug and cosmetic fixes

1.11.139 0.10.8 (2018-02-10)

- Added Plot Decay Shifts as an shortcut menu of Lifetimes Graph
- Added support for PNG file type for gate images
- Added option to not compute fit parameter uncertainties (for memory saving)
- Added separate histogram for white light image contrast
- Added ability to show overlay on white light image (and save the image)
- Added Thresholded IRF function
- Added export dataset information to Notebook right-click menu
- Added Verbose Settings Mode option
- Added Wavelet Analysis Options to Settings & Decay context menu
- Added Decay Extrapolation to Settings & Decay context menu
- Added IRF denoising to Decay Graph
- Added Selected Plots Locking to either phasor ratio references
- Added Optimal IRF Computation
- Cosmetic changes
- Added Use Cursor n as Reference n menu item
- Implemented Optimal Calibration from Reference Pair
- Implemented ROI modification
- Added Plot Editor
- Added keyboard shortcut option to New Plot dialog
- Added All ROIs Analysis script
- Allowed rearranging ROIs in ROI manager
- Added choice of individual Fit Algorithm (or Best of All)
- Output of Global Fit parameters to Lifetime Graph if requested (Displayed parameter list)
- Added AlliGator Global Fit Status Window
- Minor Bug Fixes and cosmetic improvements

1.11.140 0.9.13 (not released)

- Minor Bug Fixes (including one which would not take into account the image threshold settings)
- Phasor Ratio/Lifetime Plot now take the name of the source Phasor Plot
- Added option to only show the last decay plot
- Decay plot/fit/residuals colors can be set in Settings>>Fluorescence Decay>>Styles
- Added plots of user selected decay fit parameters during Time Series Decay Fit Analysis
- Handling of truncated decays (min and max percentile approach)
- Further improvements on lifetime fitting: added weighted fit option, parameter fit bounds, etc.

- Addition of messages for a few more script actions
- Added information output to Notebook (ROI used)
- Added Save Phasor Graph Image Option
- Added Cumulative Function extraction
- Added Source Image ROI Manager
- New Decay Plot's name is set to "Dataset Name: ROI Name"
- Delete Selected Plots function added to all Graphs
- Added Visible Items menu to all Graphs
- Implemented Global Fit of Selected Decays
- Option to export tabulated global fit results to file
- Added Selected Plot Style Dialog
- Abort option for Global Fits
- Added Average of Selected Plots
- Added Show Reference Decay/IRF
- Decay Plot Saving/Loading
- Added IRF Fit by a Gaussian-Exponential
- Support of new LaVision file format

1.11.141 0.9.12 (2017-09-08)

- Added Set/Copy Phasor Graph Cursor to/from Current Phasor
- Added ROI Statistics menu to Source Image
- Added Compute Average Plot Phasor
- Added Baseline Subtraction Method
- Implemented 1- or 2-Exp + IRF Convolution Fit (and Time-Series script)
- Added option to turn off Image Histogram/Contrast and Phasor Plot to speed up analysis
- Added Phasor (m, phi) control next to (g, s) version
- Output of Phase Lifetime each time these values are updated
- Allowed "transparent" color for ROI overlay on image or phasor plot
- Cosmetic fixes to Settings window behavior

1.11.142 0.9.11 (2017-08-22)

- Switched to LabVIEW 2017
- Added High and Low Color for Source Image and Phasor Plot
- Changed the way Image Source ROIs are stored and saved
- Introduced some optimization
- Set different file extensions for different types of XML files
- Corrected a bug reporting wrong number of files in gate folders

1.11.143 0.9.10 (2017-05-17)

- Corrected bug in Export Settings to Notebook

1.11.144 0.9.9 (2017-03-26)

- Added option to compute amplitude ratio instead of phasor ratio (Phasor Graph)
- Introduced Phasor Color Map Picker (Phasor Plot) and Phasor Color Map option (Source Image)
- Introduced Intensity Time Trace Correction option (Intensity Time Trace)
- Introduced Intensity Correction definition UI (Analysis menu)
- Added Computation of Phase Lifetime Plot option (Settings>>Phasor Graph)

1.11.145 0.9.8 (2017-01-22)

- Fixed a bug in decay background correction (min and max gate were not taken into account)
- Search ROI now returns the barycenter of all the maxima as center
- Corrected bug in Phasor Ratio plot linear fit
- Added Reject High Decay Peak Pixel options
- Minor Bug Fixes and cosmetic changes

1.11.146 0.9.7 (2017-01-13)

- Decay Smoothing and Shifting
- Recalibration of multiple phasor plots
- Saving and Loading of phasor plots
- Minor Bug Fixes and cosmetic changes

1.11.147 0.9.6 (2017-01-03)

- Improved Threshold Analysis of decays
- Improved Zero-Crossing Analysis of decays
- Analysis of decays does not require to go through IRF extraction
- Added Dataset Information Window
- Added Normalize Decay option
- Implemented cross-correlation decays shift analysis
- corrected bug with hiding tipstrip
- Optimized gate file loading
- Show File Time Stamp below image (in time series)
- Added fading Phasor Ratio Map option
- File numbering doesn't need to be justify-formatted anymore
- Phasor Ratio Reference can be defined in the Phasor Plot and using a line ROI

1.11.148 0.9.5 (2016-12-14)

- Added Source Image ROI handling options
- Added option to save the Overlay as an 8-bit image
- Introduced a Phasor Ratio Map Representation
- Various Bug Fixes and cosmetic changes

1.11.149 0.9.4 (2016-12-01, updated 2016-12-08)

- Introduced a separate Settings window, as well as saving of settings from one session to the next
- Added Multiplot capabilities in most graphs to allow comparison between different samples, etc.
- Added the ability to load a White Light Image and a ROI Mask Image
- ROI selection and handling improvements
- Added Tip strips for Plot Legends
- Various Bug Fixes and cosmetic changed

1.11.150 0.9.3 (2016-10-29)

- Added Slider in Phasor Graph panel to replay time series one step at a time
- Phasor Calibration and Phasor Calibration Curve can now be saved and reloaded
- ROIs can be saved and reloaded in both Source image and Phasor Plot image
- Actual ROI center is displayed as an overlay on the Source image
- Added option to display cumulative Phasor Plot
- Changed location of Phasor Calibration menu items

- Distinction between applying a single calibration or a curve calibration

1.12 Bibliography

The following publications have used AlliGator. Please consider citing one of them if you are using AlliGator in your work. Most of them are available for free on [PubMed](#) or [eScholarship](#)

First introduction of AlliGator (developed for this work) and demonstration of in vivo FRET analysis of time-gated data acquired with an ICCD camera. The supporting information of this article contains an extensive tutorial on phasor analysis, as well as details on data analysis with AlliGator, which can be found [here](#). Data used in this work can be found [here](#), [here](#), [here](#), [here](#), and [here](#).

Use of AlliGator for time-gated decay fitting of data acquired with a custom time-gated CMOS SPAD array (with large gates). Phasor analysis presented in this article was done with a MATLAB code.

in vitro analysis of visible range fluorescence data acquired with SwissSPAD 2, the time-gated CMOS SPAD array introduced in [Ulku2019]. This article is associated with data that can be used to test a number of features of AlliGator.

More characterization of SS2 using AlliGator, as well as MATLAB code, with associated data.

Used for some data illustrating this manuscript (but not central to the paper, which is concerned with theoretical considerations on phasor analysis)

This paper uses AlliGator for both phasor and NLSF (1-exponential and 2-exponential fits) analysis. A number of features were introduced to simplify the analysis, specifically multi-ROI, single-pixel analysis and the Fit Parameter Map, Single-exponential phasor loci (SEPL) representation, as well as significant performance improvements.

This proceedings paper uses AlliGator to analyze data where the gate width is larger than the laser period and demonstrates that phasor analysis works just as well in those cases.

This proceedings paper uses AlliGator to analyze data acquired with SwissSPAD3, which has shorter gates than the laser period and shows that phasor analysis works equally well for most cases, although the photon budget decreases for short gates.

1.13 Acknowledgments

This software was developed in LabVIEW and Python as part of research performed in the Single-Molecule Biophysics group (Weiss Lab), Department of Chemistry and Biochemistry at UCLA, in part in collaboration with the Intes Lab, Biomedical Engineering Department, Rensselaer Polytechnic Institute, Troy, NY and the Barroso Lab, Department of Molecular and Cellular Physiology, Albany Medical School, Albany, NY. It owes much to the feedback and suggestions of its users, which I hereby thank for their trust, courage and patience.

The software code includes open source code from many third party developers listed in the licenses.txt file found in the installation folder.

This work was funded in part by DOE grant DE-FC02-02ER63421-00 and DE-SC0020338, Human Frontier Science Program grant RGP0061/2015, US National Institutes of Health grants R01 GM095904 and R01 CA250636, University of California CRCC grant CRR-18-523872, and UCLA JCCC Seed grant 2020.

LabVIEW is a trademark of National Instruments.

1.14 AlliGator Support

Support for AlliGator is provided via a Google Group, located at <https://groups.google.com/forum/?hl=en#!forum/alligator-software-support>.

Anyone can consult the different topics (grouped in different categories), but group membership is required to be able to post.

1.14.1 How to report an issue

When reporting a problem, it helps if you can provide as much details as possible. In particular, try to reproduce the problem after having turned on “Verbose Error Mode” in the **Settings:>Miscellaneous** panel. If the problem is due to some bug, or failure to perform a specific task (by contrast to providing “unexpected” results), this should result in an error message displayed in red in the Notebook (**Windows:Notebook**). Please provide this error message along with your description. If you are able to share your file(s) and Notebook, this might also help with diagnosing the issue.

While it is sometimes difficult as a user to figure out whether a problem is due to a bug or data-specific, it will be helpful to everyone if issues can be classified as “cosmetic”, “question” or “bug” by posting in one of the corresponding categories.

Feature suggestions are welcome (although no warranty of any kind is made that they will be implemented) and should be categorized as such (“feature suggestion”). Other questions or comments belong to the “other posts” category.

- search

BIBLIOGRAPHY

- [Chen2018] Sez-Jade Chen, Nattawut Sinsuebphon, Alena Rudkouskaya, Margarida Barroso, Xavier Intes, Xavier Michalet, “In vitro and in vivo phasor analysis of stoichiometry and pharmacokinetics using short-lifetime near-infrared dyes and time-gated imaging”, *Journal of Biophotonics* 12 (2019) e201800185, doi: [10.1002/jbio.201800185](https://doi.org/10.1002/jbio.201800185)
- [Ulku2019] Arin Can Ulku, Claudio Bruschini, Ivan Michel Antolovic, Yung Kuo, Rinat Ankri, Shimon Weiss, Xavier Michalet, Edoardo Charbon, “A 512×512 SPAD Image Sensor With Integrated Gating for Widefield FLIM”, *IEEE Journal of Selected Topics in Quantum Electronics* 25 (2019) 6801212, doi: [10.1109/JSTQE.2018.2867439](https://doi.org/10.1109/JSTQE.2018.2867439)
- [Ankri2020] Rinat Ankri, Arkaprabha Basu, Arin Can Ulku, Claudio Bruschini, Edoardo Charbon, Shimon Weiss, Xavier Michalet, “Single-Photon, Time-Gated, Phasor-based Fluorescence Lifetime Imaging Through Highly Scattering Medium”, *ACS Photonics* 7 (2020) 68-79, doi: [10.1021/acsp Photonics.9b00874](https://doi.org/10.1021/acsp Photonics.9b00874)
- [Ulku2020] Arin Ulku, Andrei Ardelean, Michel Antolovic, Shimon Weiss, Edoardo Charbon, Claudio Bruschini, Xavier Michalet, “Wide-field time-gated SPAD imager for phasor-based FLIM applications”, *Methods and Applications in Fluorescence* 8 (2020) 024002, doi: [10.1088/2050-6120/ab6ed7](https://doi.org/10.1088/2050-6120/ab6ed7)
- [XM2021] Xavier Michalet, “Continuous and discrete phasor analysis of binned or time-gated periodic decays”, *AIP Advances* 11 (2021) 035331, doi: [10.1063/5.0027834](https://doi.org/10.1063/5.0027834)
- [JS2022] Jason T Smith, Alena Rudkouskaya, Shan Gao, Juhi M Gupta, Arin Ulku, Claudio Bruschini, Edoardo Charbon, Shimon Weiss, Margarida Barroso, Xavier Intes, Xavier Michalet, “In vitro and in vivo NIR Fluorescence Lifetime Imaging with a time-gated SPAD camera”, *Optica* 9 (2022) 532, doi: [10.1364/OP-TICA.454790](https://doi.org/10.1364/OP-TICA.454790)
- [XM2022] X Michalet, A Ulku, J Smith, C Bruschini, S Weiss, E Charbon, X Intes, “NIR fluorescence lifetime macroscopic imaging with a time-gated SPAD camera”, *Proceedings of SPIE* 11965 (2022) 1196507, doi: [10.1117/12.2607833](https://doi.org/10.1117/12.2607833)
- [XM2023] X Michalet, A Ulku, M A Wayne, C Bruschini, S Weiss, E Charbon, “NIR fluorescence lifetime macroscopic imaging with a novel time-gated SPAD camera”, *Proceedings of SPIE* 12384 (2023) 1238423