AIC iPALM Plotter GUI Manual

Jesse Aaron
Rev. Aug 2019

Introduction

The iPALM Plotter GUI is a Matlab software tool to aid in visualizing, analyzing, and filtering iPALM localization data. It does not replace PeakSelector but can add complementary capability. It was designed to take advantage of Matlab's better 3D plotting functionality, as well as the ease in computation and analysis. Importantly, this software will not render images in the way the PeakSelector will. Its main mode of displaying data is via 2D or 3D scatter plots, although a simple 2D histogram function is also available to better indicate localization density.

We recommend installing this software on a relatively recent PC/Mac with >32GB RAM and multi-core processor. To be able to run the software, you'll first need Matlab R2017b or later installed with the following toolboxes:

- **Image Processing Toolbox**
- **Statistics and Machine Learning Toolbox**
- **Curve Fitting Toolbox**

Then, you can unzip the contents of ***ipalmplotter.zip*** file into Matlab's working directory. You may choose to put those contents in their own folder within that directory but be sure to add that folder to Matlab's search path. Currently, the .zip file should contain the following files:

- **pdist2.m**
- **countEntries.m**
- **fwhm.m**
- **histogram.m**
- **ipalmcluster.fig**
- **ipalmcluster.m**
- **ipalmimport.m**
- **ipalmplotter.fig**
- **ipalmplotter.m**
- **DBSCAN.m**

To start the software, simply type 'ipalmplotter' (without quotes) at the Matlab command line.

Loading Data

Once started, the main GUI window should appear as shown in Figure 1. Currently, the software can only read ASCII .txt files generated from PeakSelector, or .txt files generated from the software itself. It cannot currently read .sav files. To load data, simply click the "Load ASCII Data" (#1 below) button in the top left corner of the window, and a file open dialogue box will open, allowing you to choose the file of interest. Depending on the size of the data, it may take some time to load all the localizations. Once it is done, the text box next to the load data button (#2) will display the current loaded data file name and location.

Filtering Data

In addition, the parameter table (#3) will populate with the first column denoting the name of each parameter calculated for each localization, and the minimum and maximum value for each parameter in columns 2 and 3, respectively. This table is editable such that you can further filter your data if you choose to. Simply click on the min/max value you wish to change, enter a new value, and then press the **Filter Peaks** button below (#4). At any point, you can press the **Reload Limits** button (#5) to return the table to its original min/max values. You can view the total number of filtered localizations in the box titled "**Num. Particles**" (#6).

2D or 3D Plotting of Data

You can plot any of the parameters listed in the table vs. any other parameter in the large axes on the right side of the gui (#7). To select which parameters are to be plotted on up to 3 axes, use the drop down menus denoted in (#8). Then, press the **PLOT DATA** button (#9). Most often, users will want to plot the x vs. y vs. Unwrapped Z (or their "group" equivalents). For these common scenarios, there are "shortcut" buttons, denoted by (#10). In other cases, you may only want to make a 2D plot of 2 parameters. In such cases, you can select "none" in the **Z Axis Values** dropdown menu, denoted in (#8). You can adjust the appearance of the of the plot via the the menu items on top the plot axes, denoted by (#11-14). The **color by** (#11) menu allows you to add a color to each point in the scatter plot to indicate a relevant parameter: select "none" for all points to be black, select "label" to color each point according to the color channel it was acquired in (red = 647nm excitation, green = 561nm excitation, and blue = 488nm excitation). Or, select "Z axis" value to color each point according to the value specified in the **Z Axis Values** dropdown menu. **Marker Style** (#12) allows you to change the symbol used in the scatter plot, while **Marker Size** (#13) controls its size. Finally, you can adjust the aspect ratio of the axes such that they represent the same scale by clicking **make axes equal scale** (#14). This is particularly useful when plotting x vs. y vs. z (or unwrapped z) values.

To further manipulate plots, there are a number of options. To zoom in on a subregion of the plot, you can click **Zoom (Rectangle)** (#15) or **Zoom (Polygon)** (#16). The former will prompt you to draw a box around the region of interest, the later will allow you to draw an arbitrary polygon if needed to better isolate a structure of interest. Clicking **Unzoom** (#17) will return the plot to its previous xyz limits. You can view the scatterplot in standard ortho views. Clicking the **XY** (#18), **XZ** (#19), and **YZ** (#20) buttons will rotate the axes to show the projections as indicated. You can also click the data tip tool on the upper left of the GUI (#21), which will display the x, y, (and z if applicable) value for any given point in the scatter plot, as well as freely rotate the axes (#22) at any angle.

Histograms

This distribution of each parameter in an iPALM data set can be plotted in the axes located in the middle of the gui (#23). Choose the parameter to be plotted via the dropdown menu indicated by (#24), then press the **Plot Histogram button** (#25). Some parameters are best viewed with one or more axes in log scale, so you can adjust the plot accordingly with the **X log scale** (#26) and **Y log scale** (#27) check boxes. Furthermore, you can fit a distribution to a Gaussian function by clicking the appropriate check box (#28). This is especially useful for measuring xyz distribution sizes. Summary statistics for the histogram are shown in the table below (#29). They include the peak bin location, mean, median, and FWHM (if calculable). If the **Fit to Gaussian** check box is selected, the Gaussian amplitude, mean, and sigma value will also be displayed in the table. If a dataset contains more than one label, each summary statistic will be calculated separately for each color, and displayed in separate columns as indicated. Finally, the histogram data (bin locations and frequencies) can be exported into a .csv file by clicking the **Export Histogram** button (#30). ***Note that the histrogram considers only those points visible in the scatterplot, so any zooming and/or filtering will change the histogram.***

Analysis

A few analysis algorithms have been added to the GUI, with others currently in development. Currently, there are 3 point-wise analysis methods that are available. These are all listed at the far right of the GUI under the **Point Analysis** panel (#30).

DBSCAN

Short for Density-based spatial clustering of applications with noise, this is a common method of grouping localizations into "clusters". There are a number of publications that outline its use, assumptions, e.g.:

Sander, J., Ester, M., Kriegel, HP. et al. Data Mining and Knowledge Discovery (1998) 2: 169. https://doi.org/10.1023/A:1009745219419

The algorithm identifies localizations based on their local density, and is advantageous as it does not bias those clusters based on shapes. It can efficiently group localizations as "clustered" or "noise" based on this local density. Clicking the **DBSCAN** button prompts the user for three parameters:

Eps: the search radius when looking for other particles
Min. Points: The minimum number of neighbors that must constitute a cluster
Color Channel: The color channel to be considered (1 = red, 2 = green, 3 = blue).

The GUI will then perform DBSCAN, and assign each peak in the 2D/3D plot. **As with the histogram, DBSCAN only operates on the points shown in the scatterplot, so zooming/filtering will affect the results.** After DBSCAN finishes (which may take a while depending on the number of points), each currently visible point will be given a "cluster" value in the table shown in (#3). A value of zero indicates that point is not part of a cluster, while points within a cluster are given a number unique to that cluster. Therefore, you can filter your data such that only clustered points are shown by typing a minimum cluster value of >0 in the filter table. Or, only specific cluster(s) can be viewed by likewise changing the filter settings in the table.

Colocalization

A method published by Malkusch et al. proposed a point-wise Pearson's correlation coefficient approach to measure co-localization in multi-channel single molecule localization data sets:

Malkusch, S., Endesfelder, U., Mondry, J., Gelléri, M., Verveer, P. J., & Heilemann, M. (2012). Coordinate-based colocalization analysis of single-molecule localization microscopy data. *Histochemistry and cell biology*, *137*(1), 1-10.

In this method, a coordinate based colocalization (CBC) value is calculated between color channel 1 and channel 2, and/or vice versa. These values can range from -1 to +1, indicating strong anti-correlation, and strong correlation, respectively.

The GUI will perform CBC calculations between green and red or between red and green channels (support for 3 color is upcoming). Pressing the **Colocalization** button will prompt the user for 3 parameters: (1) the direction (1 = colocalization of red to green, 2 = colocalization of green to red). Note that this analysis can be performed twice to calculate the "total" colocalization for each channel. (2) A max search radius is needed to calculate the distribution of neighbors – this is typically the maximum size of the features of interest. And (3), a search increment. Typical values are ~1% of the max search radius (e.g. 100nm search radius, 1nm increment).

After running the algorithm, the GUI then displays localizations that are color coded for their CBC scores.

A colocalization value is assigned to each localization (depending on the direction parameter), and can then be filtered using the table indicated by (#3) and button in (#4) to, for example, only show localizations with CBC values of greater than a chosen threshold.

Nearest Neighbor
The GUI will also calculate a simple nearest neighbor distance for each localization. Pressing the **Nearest Neighbor** button will prompt the user to specify color channel **from** and color channel **to** distance. For example, if the user selects 1 and 2, respectively, the GUI will calculate the nearest neighbor (NN) distance from each color 1 localization to its respective color 2 localization. This can be used as a simpler proxy for colocalization. Alternatively, if the user selects the same **to** and **from** color channel, this can give complementary information on cluster size and/or can be used to remove outliers. Note that a user can select complementary to and from color channels. For example, the NN calculation can be done from channel 1 to channel 2, and repeated but from channel 2 to channel 1. Doing so will assign NN distances to both channels. Similarly, choosing channel 1 for both to and from (or channel 2) will result in unique NN distances for each channel.

As with the previous analysis, nearest neighbor distance can be used as a basis for filtering your data in the filter table (#3) to, for example, only show localizations of one channel that are within a threshold distance to localizations in another channel.

Exporting Analysis Results
The results from any of the above analysis methods are stored as additional parameters within the localization data. To export these results, in addition to the original localization data, press the **Export Filtered/Analyzed Points** button (#31). Doing so will export current, filtered data to a new tab-delimited ASCII text file, similar to what PeakSelector will produce. This data can be re-loaded into the GUI for further analysis if necessary, or analyzed via other analysis software. The GUI will first prompt for a physical pixel size – for iPALM data, use the default 133nm to save x/y positions in pixels, or use 1 if you want to save xy positions in terms of nm (this will prevent the data from being correctly re-imported into PeakSelector, however). A save file dialog will open, prompting the user to input a new file name.

Global Cluster Analysis
Using DBSCAN to identify clusters may only be the first step in analysis. To probe further, there is a **Cluster Analysis** button under the **Global Analysis** button group (#32). Clicking this button will open a new window, shown in Figure 2. Similar to the main GUI, there is a Load Analyzed Data (.txt) button at the top left (#33). This will again prompt the user for a pixel size. If using iPALM data, you can use the default 133nm/pixel value. You can then select a .txt file. Importantly, this file should be one that was saved using the main GUI, and that contains the results from DBSCAN analysis. Once loaded, the name of the file selected will be displayed in the text box in (#34)

Once the data is imported, you can plot the data in one of three ways. At the bottom of the axes at left (#35), there is a dropdown menu (#36). You can choose to plot all localizations, only those localizations that are members of a cluster, or only those localizations that are NOT a member of a cluster. Once you choose the display mode, you can click **Plot Data** Button (#37). Similar to the main GUI, you can rotate the data using the button shown via (#38). **Zoom functionality will be added at a later date.**

Once data is loaded, you can push the **Calculate Cluster Parameters** button (#39). This will prompt Matlab to take all the points in each cluster (and in each label containing cluster indices), and attempt to create an Alpha Shape from this data. You can read more about alpha shapes here: http://wcl.cs.rpi.edu/papers/b11.pdf . This method offers a way to create shape descriptors given a set of points that occupies a finite sized cluster. Currently, the algorithm calculates **8 parameters** for each cluster, including:

- Number of points in the cluster
- The cluster centroid x-position
- The cluster centroid y-position
- The cluster centroid z-position
- Volume
- Surface Area
- Surface Area to Volume Ratio
- Effective Diameter (assuming spherical geometry)

This calculation may take some time, depending on the number of clusters present in the data. A summary table is then automatically populated for each label (color channel) in the data, which is summarized in (#40-42). For each label, the tables show the minimum and maximum values of each of the eight parameters listed above. Only labels that both exist, and have cluster indices associated with them will be summarized in their respective table.

Clusters can now be filtered based on these 8 parameters. To do so, simply type a new number for the min or max value for any of the parameters (and for any of the applicable labels). Then, press **Filter and Re-Plot** (#43). The axes in (#35) will then be updated to show only clusters that satisfy the filtering specifications. You can always re-load the original data limits by pressing the **Re-Load Limits** button (#44), which will also update the table(s) to their original min/max values.

You can also plot any of the 8 parameters above as histograms (for each label). To do so, first select the parameter to be displayed from the dropdown menu in (#45). You can then push the **Plot Histogram** button (#46), which will display on the axes in (#47). It may be useful to plot the x and/or y axes in log scale in some instances. In this case, you can click either of the check boxes titled **Log X** or **Log Y** (#48).

You can also compute the distances between cluster centers or peak localizations of one label to another (or among the same label). To do so, select the appropriate “from” and “to” parameters from the dropdown menus shown by (#49) and (#50), respectively, and then push the “**Compute Distance**” button (#51). For example, choosing:

- “From” label 1 clusters “To” label 2 clusters, will result in the distribution of nearest neighbor distances from the centroids of all label 1 clusters to the nearest label 2 cluster centroid.
- “From” label 1 clusters “To” label 2 peaks, will result in the distribution of nearest neighbor distances from the centroids of all label 1 clusters to the nearest label 2 peak.
- “From” label 1 peaks “To” label 2 peaks, will result in the distribution of nearest neighbor distances from all label 1 peaks to the nearest label 2 peak.
- Etc.

The distribution of selected nearest neighbor distance metric will be displayed in the axes denoted by (#52). Finally, the results of the global cluster analysis can be saved as a Matlab .mat file by pressing the **Save Results** button (#53). These files can be directly imported into Matlab for further analysis or visualization, depending on the application.

The data is saved as a matlab structured array, with the default name of “output”. To load the saved .mat file into matlab, you can simply drag the file into the Matlab workspace (it may take some time depending on the size). A variable with name “output” should appear when done loading. Typing the variable name in the command prompt will give a result similar to the one shown below:

```
      peaksdata: [202597×5 double]
         Label1: [1×1 struct]
         Label2: [1×1 struct]
         Label3: [1×1 struct]
     NNdistFrom: {'Label 1 Clusters'}
       NNdistTo: {'Label 2 Clusters'}
    NNdistances: [3676×1 double]
```

This represents the “first level” data organization. For example, typing :

```
output.peaksdata
```

Will return the original data, with peaks in each row, and each column corresponding to x, y, z-location, color channel, and cluster index, respectively. Typing:

```
output.NNdistFrom
```

Will return the choice made in pull down menu (#49). Likewise typing:

```
output.NNdistTo
```

Will return the choice made in pull down menu (#50). Typing:

```
output.NNdistances
```

Will return the nearest neighbor distances between the “From” and “To” parameters listed above.

Now, typing

```
output.Label1
```

Will give a result similar to this:

```
        clusterprops: [3676×8 double]
              shapes: {3676×1 cell}
      parameternames: {8×1 cell}
        paramfilters: [8×2 double]
    filteredclusters: [3676×8 double]
      filteredshapes: {3676×1 cell}
```

This represents the “second” level data organization, and is particular to each label that is present. So, for example, if we want to view each of the 8 cluster properties for all label1 clusters, type:

```
output.Label1.clusterprops
```

This will return a matrix, with each cluster represented by row, and each of the 8 cluster parameters by column. The other parameters are as follows:

- **shapes**: cell array of each alpha shape corresponding to each cluster. This is, in turn, another structured array. Contact the AIC for more details here if needed.
- **parameternames:** cell array containing descriptions of each of the 8 cluster parameters
- **paramfilters**: the min/max parameter values used to filter clusters
- **filteredclusters**: similar to clusterprops, but with the parameter filters applied
- **filteredshapes**: similar to shapes, but with the parameter filters applied

This software is under active development, and updates will be available as they are made. Of course, if you have any questions or run into issues, please don't hesitate to contact AIC staff.

Figure 1: Main GUI

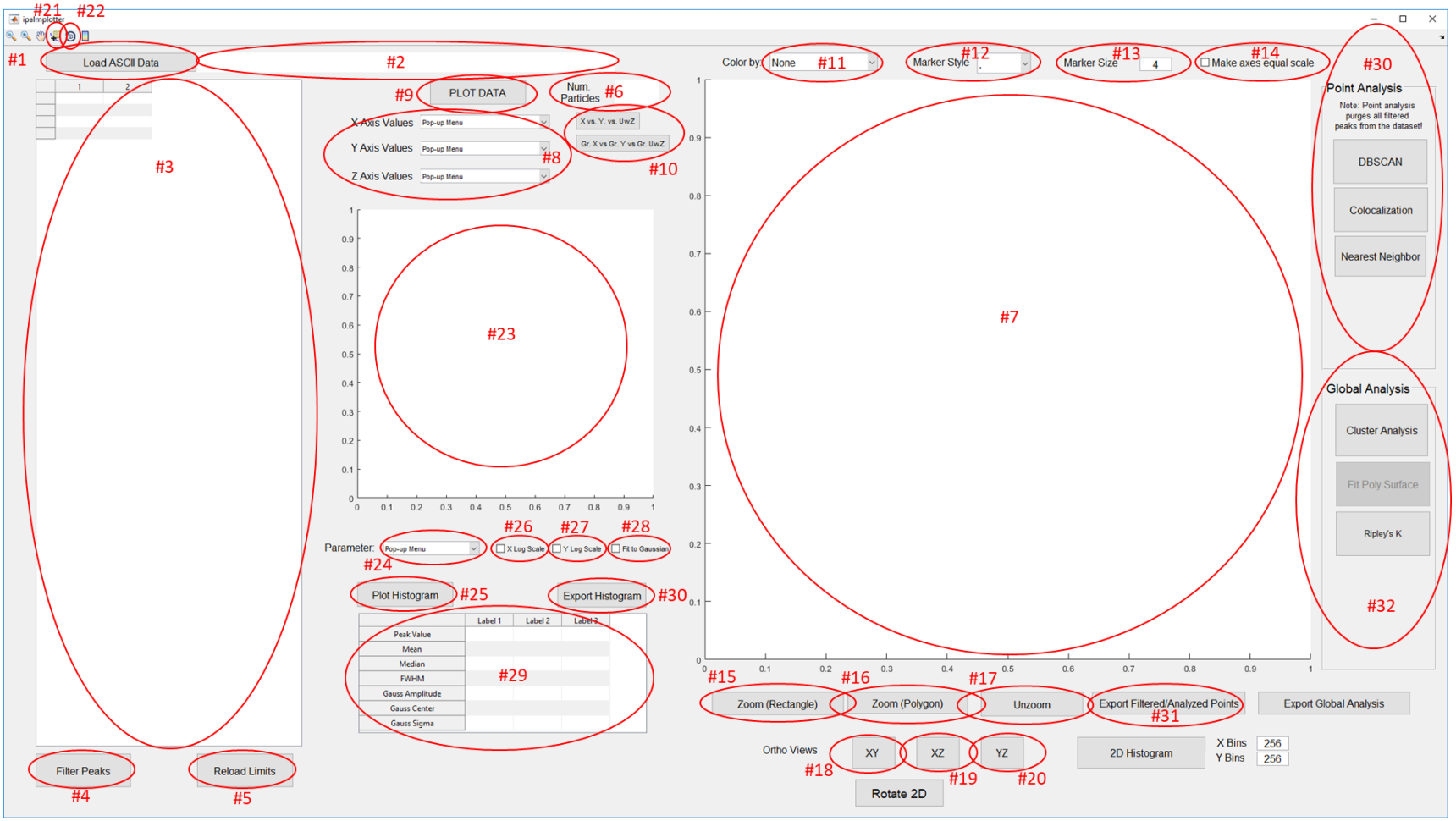

Figure 2: Global Cluster Analysis