SARFIA user manual

Mario M. Dorostkar
MRC Laboratory of Molecular Biology
Hills Road
Cambridge
CB2 0QH UK
mmd@mrc-lmb.cam.ac.uk

Contents

1 System requirements

SARFIA requires Igor Pro 6.1 or later, running either Windows XP (or later) or Mac OS X 10.4 (or later) . The latest version of Igor Pro can be downloaded from `http://wavemetrics.com/`. SARFIA is available free of charge from `http://www.igorexchange.com/project/SARFIA` under an academic free license.

2 Installation

The SARFIA package comes packed in a .zip file, containing subfolders. Extract the files into the appropriate subfolders in the `MyDocuments\WaveMetrics\IgorPro6UserFiles` (Win) or `Documents:WaveMetrics:IgorPro6UserFiles` (Mac) folders. The files can be placed into subfolders in these directories. Restart Igor Pro. If the SARFIA help file was placed in the help files directory, it will be compiled when Igor starts for the first time. The SARFIA help file contains detailed information on all functions supplied with SARFIA.

After starting Igor Pro, a new menu item, called "SARFIA" should appear (Fig. 1). From this, control panels are launched and the most important functions can be called.

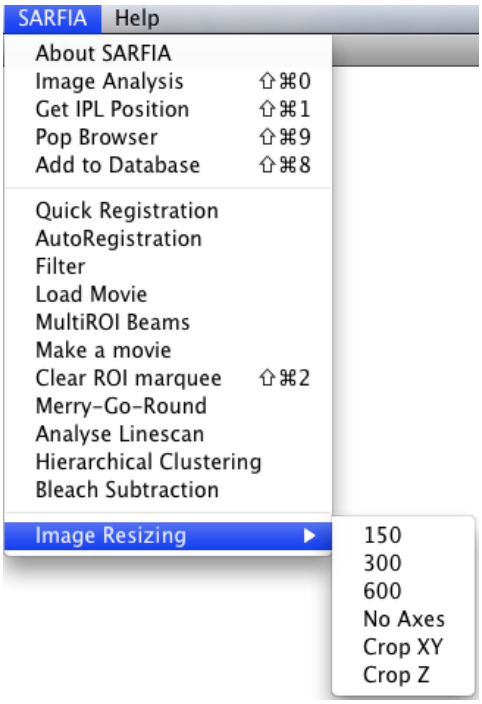

Fig. 1: The SARFIA quick start menu.

3 Data import

Tiff images (1, 8, 16 or 24 bit) can be loaded using the **Load Image** button. This will load a single file which can either be a single image, an image stack or a 24 bit RGB image, and store it in a 2D or 3D wave. Furthermore, numbered tiff files can be imported into a 3D wave by calling **Load Movie** from the SARFIA menu. In that case the file names must follow the following format: `*nnn.tif` with `n` being digits. Select the first file, and all files with increasing numbers present in the folder will be loaded into the stack.

Once loaded, images are saved as single precision waves. Most SARFIA operations, with the exception of image registration, also work on double precision waves, but not on integer waves. Thus, manually loaded waves have to be redimensioned using the `redimension /s` or `/d` command in order to produce meaningful results.

4 Quick Analysis – extracting time-series information

4.1 Image Analysis control panel

The Image Analysis control panel (Fig. 2) allows access to general image manipulation procedures, thresholding and extraction of time-series data.

Load Image loads a .tiff file into Igor. If the file was acquired with ScanImage3, automatic scaling is applied to the image. However, in order for the scaling to be correct, two constants in the file `LoadScanImage.ipf` have to be modified (Fig. 3): Z_factor calibrates the z value of "1μm" in ScanImage to a real world distance. In our case, the setting of "1μm" in SanImage corresponded to real-life value of about 4μm. ImageLength specifies the width and height of a square image in μm recorded at a zoom factor of 1. **Note**: The scaling of these images in Igor is in m and s. Igor automatically applies the correct prefix, i.e. μ when displaying a wave.

If images were acquired a separate files, i.e. using the loop command in ScanImage, the function `LoadMovie()`, which can be called from the SARFIA menu, can be used to load consecutively numbered image files into a single 3D wave.

ROI Panel opens the inbuilt panel to draw a region of interest (ROI) on the top graph. The result is stored as M_ROImask.

Contrast launches the inbuilt control panel to adjust contrast in the top graph. This works only on 2D waves. **Note**: This will change the actual values of your data. To change the contrast in a graph, right-click the graph, then select "Modify Image Appearance..." and change the first color/ last color settings. These will change the way the graph is displayed, but not the data.

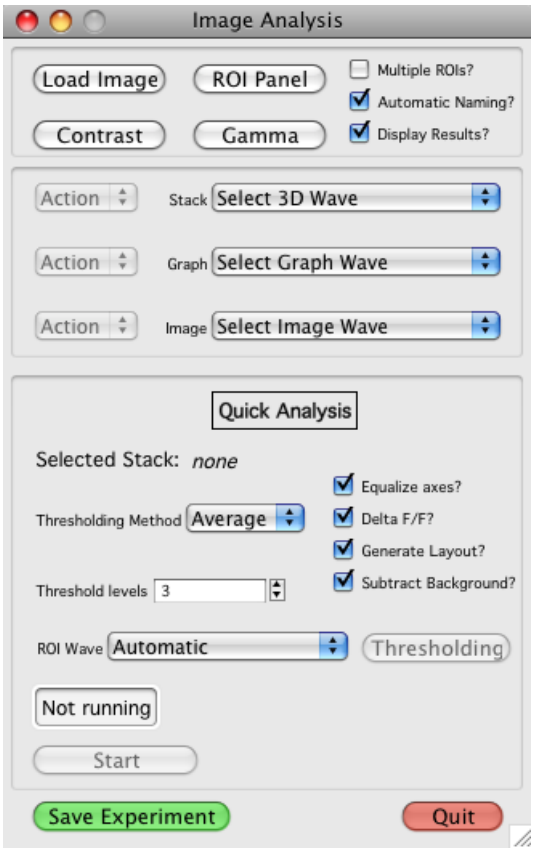

Fig. 2: The image analysis control panel.

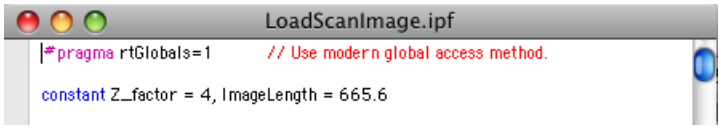

Fig. 3: Calibrating the automated scaling for ScanImage files.

Gamma launches a control panel to change the gamma value of the top graph. This function will generate a lookup table for the graph and thus does not change the data. **Note**: Changing the gamma value of an image constitutes a non-linear adjustment that has to be disclosed in most scientific journals.

The checkbox **Multiple ROIs?** specifies whether the manually drawn ROI mask should be treated as multiple ROIs. This has only an effect when performing the "Z-Stack" command on a stack.

The checkbox **Automatic Naming?** specifies whether results will have automatically generated names. If this is unchecked, you will have to specify a name each time a function is called that produces a new wave.

The checkbox **Display Results?** specifies whether results of functions called via the control panel are automatically displayed.

The popup menus **Select 3D Wave**, **Select Graph Wave**, and **Select Image Wave** provide lists of the appropriate waves in the current data folder. Once a wave is chosen, the **Action** menu to the left becomes active and lets the user choose different operations to perform on the currently selected wave.

The **QuickAnalysis** controls provide access to the functions to retrieve data from the z domain of an image stack. To analyse a stack, first select a 3D wave from the popup menu. Then, select a thresholding method and click the button **Threshold** (see Subsection 4.2). Alternatively, a previously generated ROI wave can be specified from the popup menu **ROI wave**. Once thresholding is completed, click **Start** to start the analysis.

The checkbox **Equalize Axes?** specifies whether all axes in the quick analysis layout are equalised. This is useful for comparing responses, but is susceptible to outliers. The checkbox **Delta F/F?** specifies whether the normalised or raw fluorescence results are displayed in the quick analysis layout. Both are calculated in either case. The checkbox **Generate Layout?** specifies whether a quick analysis layout is automatically generated. The first two checkboxes have no effect, if **Generate Layout?** is unchecked. The checkbox **Subtract Background?** specifies whether background subtraction will be done on the data. If it is checked, the a popup will appear when **Start** is clicked that lets the user specify a background area. Alternatively, a background corrected wave could be selected as from the popup menu.

The button **Save Experiment** saves the experiment under the name of the currently selected 3D wave, if it hasn't been saved before. Otherwise, the experiment will be saved under the current name.

The button **Quit** closes the Image Analysis control panel and removes its associated global variables.

4.2 Thresholding

Thresholding is performed on the Laplace operator of the selected image or stack, or alternatively on the à trous wavelet transform. The latter is described in detail in Olivo-Marin, 2002, Pattern Recognition 35 (9), 1989 – 1996. In order to threshold the raw brightness, use the ImageThreshold function, which is accessible from the in-built image processing functions. These can be launched from the Igor menu from Analysis>Packages>Image Processing.

The Thresholding control panel (Fig. 4) is launched either from the **Action** menus or, if the **Thresholding** button is clicked. The control panel appears together with a preview window. The variable **Threshold levels** sets the threshold in negative standard deviations of the Laplace operator of the image. The second variable, **Remove ROIs smaller than (px)** sets the minimum threshold in pixels for ROIs. The buttons **Contrast** and **Range** call the inbuilt control panels for image contrast and image range on the preview image. The button **Save Setting** saves the ROI mask as MTROIWave. **Note** that any previous wave MTROIWave will be overwritten. Duplicate or rename the original to keep it. **Cancel** cancels without saving a ROI mask.

When thresholding based on the à trous wavelet transform, two more variables can be adjusted: The level of correlation, J, which defines the spatial scale at which features are filtered. Note that only even values will produce meaningful results. The filter value, k, defines a threshold, below which wavelet coefficients are ignored. Furthermore, k is scaled by the median absolute deviation of the wavelet coefficients. Thresholding based on the à trous wavelet transform may be superior for the detection of small (i.e. pixel-sized) spots in an image.

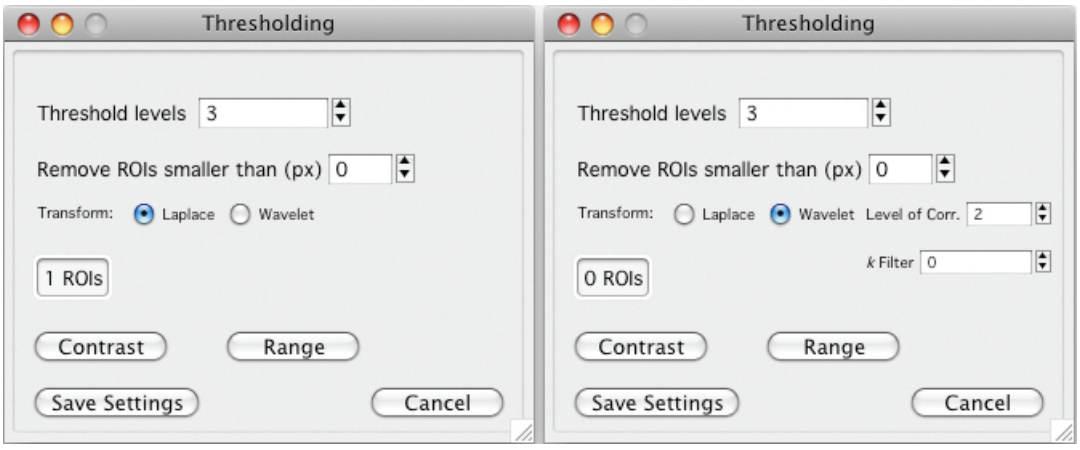

Fig. 4: The Thresholding control panel. Left: Controls for thresholding based on the Laplace operator. Right: Controls for thresholding based on the à trous wavelet transform.

4.3 MultiROI Mask

The results of the thresholding operation are saved as a MultiROI mask. This is a 2D wave, which stores the identity of each ROI as a negative number, starting with −1 Thus, ROI #0 has the value −1, ROI #1 −2 and so on. Pixels can be removed from a mask by first drawing a marquee on the graph and then selecting "Clear ROI marquee" from the SARFIA menu. This also re-runs the segmentation and numbering algorithm. Thus, if ROIs were split in the process, the resulting ROIs will be given different numbers.

Note: A standard binary ROI mask can be segmented and numbered using the following command:

```
multiROI(M_ROIMask, "MultiROIMask")
```

This function uses an iterative flood-filling algorithm to segment the ROIs (pixels with the value 0) in the wave M_ROIMask and assigns them consecutive negative numbers. The result will be a double-precision floating point wave, regardless of the type of the input wave. In this example, the result would be named MultiROIMask.

4.4 Extracting the time-course for all ROI pixels and clustering pixels based on their time-course

The time-courses for all pixels in a binary ROI mask (note that pixels in the ROI have a value of 0, those outside, 1) or for all pixels of all ROIs in a MultiROI mask can be extracted by choosing **MultiROI Beams** from the SARFIA menu. In the latter case, an index wave will point to the ROI number the traces originated from. The beams will be stored in a populationwave, which can be used as an input for the hierarchical clustering algorithms.

Proper time-courses for each ROI can be calculated by using the following command:

```
ROIBeams2Traces(ROIBeams, index, ResultName)
```

Note that *index* can be replaced by a suitable category wave, for instance the wave *Clusters*, which is the output of the SARFIA clustering algorithms.

4.5 Determining which centres of mass coincide with regions of interest — Analysing overlays

The function `CoMbyROI(CoM, ROI)` calculates the centres of mass, as specified by *CoM*, which coincide with a region of interest, as specified by *ROI* (Fig. 5). The result is stored in *w_CoMbyROI* (Fig. 6), where the row indicates the index of the centre of mass and the entry at that row specifies the ROI on which it lies, or NaN if it is not on any ROI. These results can then be processed by making a histogram with the same number

of bins as there are ROIs in ROI. This histogram will then whow the number of centres of mass for each ROI.

A control panel can be launched by typing `CoMByROICP()`. This prompts for the 2 waves CoM and ROI and allows the user to enter a value (in pixels) by which the regions of interest can be dilated before the analysis in order to detect centres of mass on the border of a ROI.

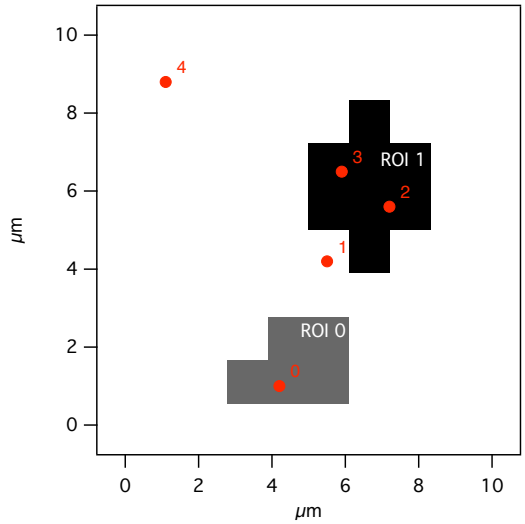

Fig. 5: Overlay showing centres of mass (red spots) on regions of interest (black and gray). These can be analysed with the function CoMbyROI.

RO 0

Point	w_CoMbyROI
0	0
1	
2	1
3	1
4	
5	

Fig. 6: Results of the function CoMbyROI of the data shown in Fig. 5.

5 PopWave Browser

Time-series results are generally stored in a so-called populationwave. This is a 2 dimensional wave, that stores single traces in its columns. The **Pop Browser** (Fig. 7)

can be used to display traces from a populationwave or a database (see also section 14). Furthermore, a second populationwave (i.e. results of a curvefit) or a stimulus can be overlaid.

The function `PopX2Traces(PopWave,BaseName)` , which can be called from the Image> Action menu of the Image Analysis Control Pane splits a populationwave into its traces. However, this produces a large number of waves, and many analysis function rely on populationwaves as input. Therefore, splitting populationwaves is generally not recommended.

The popup menu **PoP** specifies the populationwave to be displayed. **Index** is used to set the index of the trace currently shown. To its right, the wave and subrange currently displayed are shown. The button **To Clip** copies this string to the clipboard. The forward and backward buttons advance/diminish the index by one.

The second popup menu, **Stim/Cat**, specifies a second wave to be displayed. The radio buttons **Stim** and **Pop/DB** specify whether this second wave shows a stimulus (see below) or a second populationwave or database. The difference is that a stimulus will be displayed on the right y axis, and the same stimulus will be displayed for any index. If a second populationwave or database is displayed, it is assumed to have the same number of traces, and traces of the same index are both displayed on the left y axis.

The checkbox **Fix y-Axis** fixes minimum and maximum of the currently displayed y axis when the index is changed or a different wave is selected. When unchecked, the display is automatically updated. **Tip**: You can change the range of the y axis manually by double-clicking on the y axis in the graph. This opens the standard Igor control panel to adjust the axis. The same is true for the graph itself or the x axis. Settings other than the y axis range, however, are reset when the index is changed or a different wave is selected. The checkbox **Log right axis?** changes the right axis scale from linear to logarithmic. Zero values will not be shown on a logarithmic scale.

Making a stimulus wave:
A stimulus wave has 2 columns and any number of rows. The rows hold the time-series information, the first column is displayed in amber, the second in blue. If either is not needed, the values can be set to NaN, so that they won't be displayed. For instance, the stimulus displayed in Figure 8 was generated in the following way:

```
make/o/n=(300,2) stim = 0
setscale /i x,0,30,"s" stim
stim(5,10)[0]=1
stim(25,27)[1]=0.5
```

The actual number of points in the x dimension have to be sufficient to encode the maximum frequency used. In the above example, 30 would have sufficed, since the stimulus durations were in whole seconds. To display a stimulus from 5 to 7.5 s, for instance, at least 60 points would be necessary. The y dimension has to have exactly 2 columns so that the wave is correctly recognised as a stimulus.

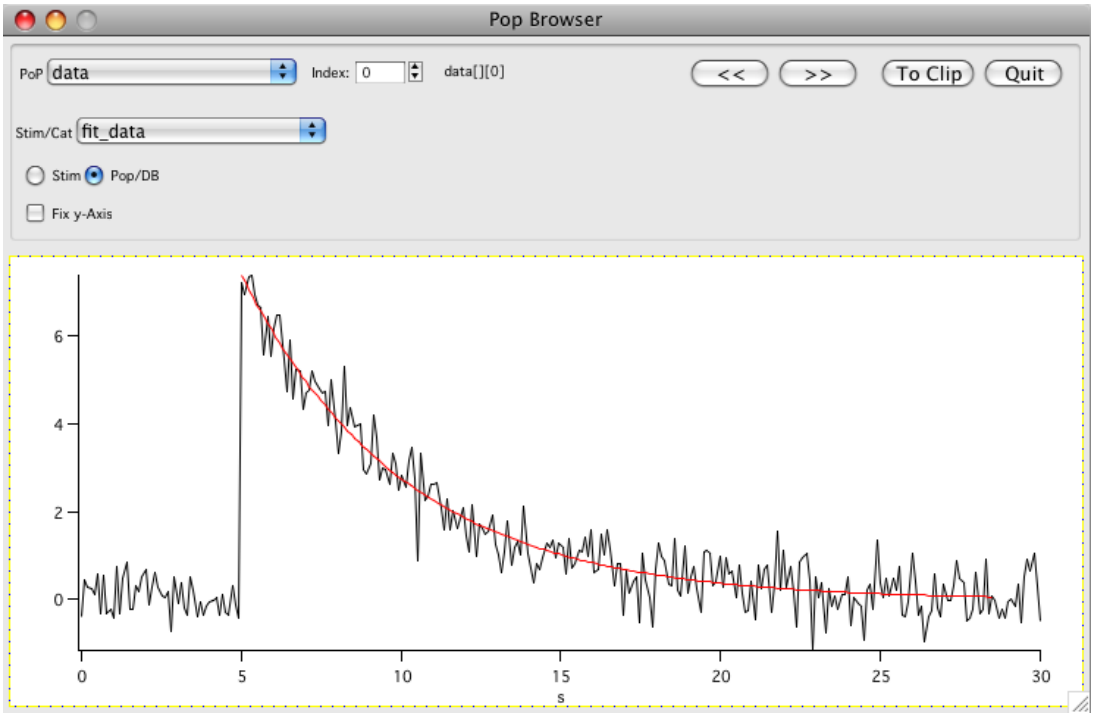

Fig. 7: The populationwave browser displaying a trace from the populationwave data (black), and the results of a curvefit (red).

Tip: If such a stimulus wave, named "stim" or "train", is present in the current data folder, when the quick analysis is started from the Image Analysis control panel, then it will be displayed in the quick analysis layouts. "Train" will be better displayed for high-frequency stimuli.

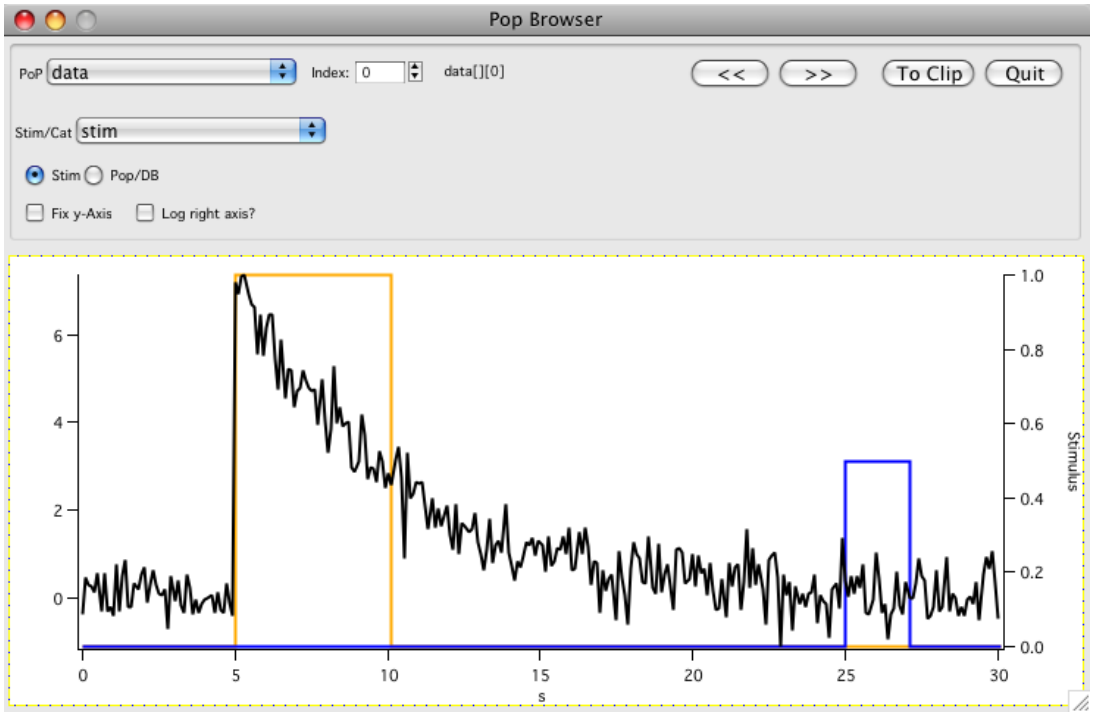

Fig. 8: The populationwave browser displaying a trace from the populationwave data (black), and the stimulus (amber and blue).

6 Rotating images

Images and stacks can be rotated in the x-y plane by selecting "Merry-Go-Round" from the SARFIA menu or by choosing the appropriate command from the Image Analysis control panel. Either way, the Rotate Image control panel (Fig. 9) will be called.

Matrix Rotation will call a custom-written function that rotates an image by assigning the value of a pixel in the original image to the closest pixel in the resulting image. **Image Rotation**, in contrast, performs interpolation, i.e. spreads the value of a pixel in the original over several pixels in the resulting image (see Fig. 10). Matrix rotation is useful if pixel values have to be preserved, e.g. if a ROI mask is being rotated. **Scaled image rotation?** invokes preservation of image scaling for the image rotation algorithm. This has an effect only, if the angle is a multiple of 90 °.

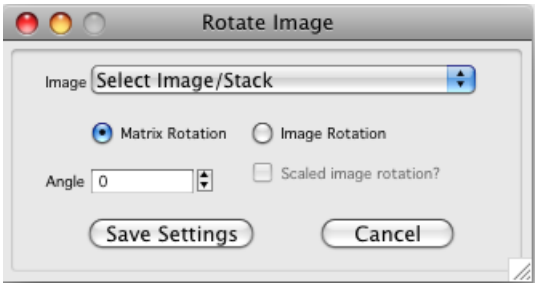

Fig. 9: The Rotate Image control panel.

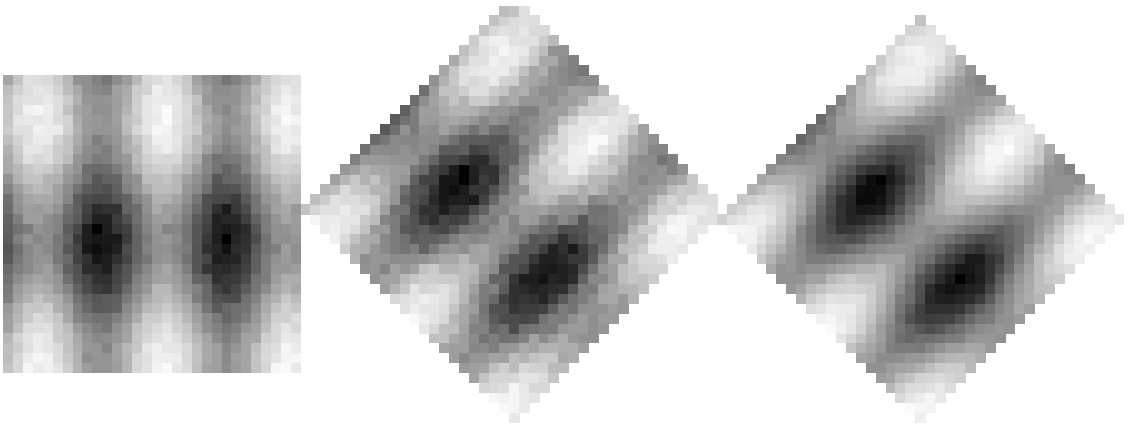

Fig. 10: Image rotation using the matrix rotation (assigning pixel values to new locations) and image rotation (with interpolation) algorithms.

7 Cropping images

The most flexible way to crop image data is using the inbuilt `duplicate` command with the `/R` flag, eg.

```
duplicate/o/r=[x1,x2][y1,y2][yz1,z1] ImageStack, CroppedStack
```

crops the imagestack to the pixels specified by the variables `x1, x2, y1, y2, yz1, z1` and stores the result in `CroppedStack`. In order to include a whole dimension, enter `[0,*]`. In order to crop by scaled variables rather than points, use round brackets. More details can be found in the Igor manual.

Users who prefer to use point-and-click methods can use the following method: Images and stacks can be cropped in the x/y plane by drawing a marquee on a graph of that image or stack and then selecting Crop XY from the SARFIA>Image Resizing menu. Cropping in the z axis is done by selecting Crop Z from the SARFIA>Image Resizing menu, and then specifying the frames or scaled variables.

8 Image registration

Image registration is used to correct for motion artefacts in image stacks. Igor supports offset, rotation, scaling and skewing. Igor's inbuilt `ImageRegistration` function can be accessed from the Image Analysis control panel's action menu, or from the SARFIA menu. The latter offers 2 options, **Quick Registration**, which performs registration on the top graph (provided it shows a 3D wave), and overwrites it with the result (the function accessed from the Image Analysis control panel creates a copy). **AutoRegistration** lets the user specify any number of files in a folder on the computer's hard drive. These will be sequentially loaded, registered, saved in the same folder as Igor Binary (.ibw) files and then removed from the memory. This can be used to register a large number of stacks over night or while away from the computer.

9 Filtering images

Filtering of images or stacks can be used to smooth or de-noise images or image stacks. Igor provides the following types of filters for images, image stacks and matrices: average, Gauss, hybridmedian (3D) or find edges (2D), max, median, min, point; These are described in the Igor Pro documentation. Furthermore, smoothing by principal component analysis (PCA) is implemented.

Smoothing by **PCA** calculates the principal components from a 3D wave. Then all components higher than the number specified in the control panel will be set to 0 and a smoothed data set will be reconstructed from the remaining principal components.

Igor calculates as many principal components as the image has pixels, therefore high-resolution images may run slow or overburden the computer's memory.

The **Filter parameters** control panel (Fig. 11) lets the user select the method from a popup menu. Note that on 2D waves, the find edges method will be applied when Hybridmedian is selected. **Filter size** specifies the filter size in pixels. A larger filter size means stronger filtering. In most situations it will be useful to set it to an odd number in order to preserve the symmetry in the filters. If PCA is selected, however, this field will set the **number of principal components** that the smoothed image stack will be reconstructed from. A larger number of principal components means less filtering. **Filter in z-axis** specifies whether filters will be applied in three dimensions or applied separately to each frame of of the stack. This setting has no effect, if the specified data has 2 dimensions, or if PCA is chosen as a method.

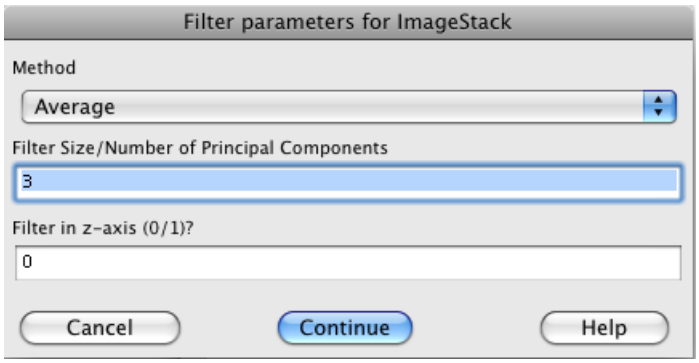

Fig. 11: The Filter parameters control panel.

10 Resizing graphs and images

In order to display graphs with the correct aspect ratio of axes, use the `SizeImage(Size, [WindowName])` function. This function corrects the aspect ratio, even if the image doesn't fill the entire graph. Size is the length of the longer side of the graph, and the optional parameter WindowName specifies the graph window. If left out, the command will be applied to the top graph. This function can also be accessed from the SARFIA>Image Resizing menu for a few pre-set sizes.

The menu item **No axes** from the SARFIA>Image Resizing menu removes the axes from a graph, which is usually desired when making graphs from iages for presentations or publications.

11 Analysis of linescan data

Linescan data is acquired by repeatedly scanning the same line in a sample with a laser-scanning microscope. These data are saved as an image, where the lines in the image represent the different time-points or as stacks of such images (Fig. 12). Such data can be analysed by selecting **Analyse Linescan** from the SARFIA menu. This opens a control panel (Fig. 12), from which the user specifies the left and right pixel of a region of interest in the data.

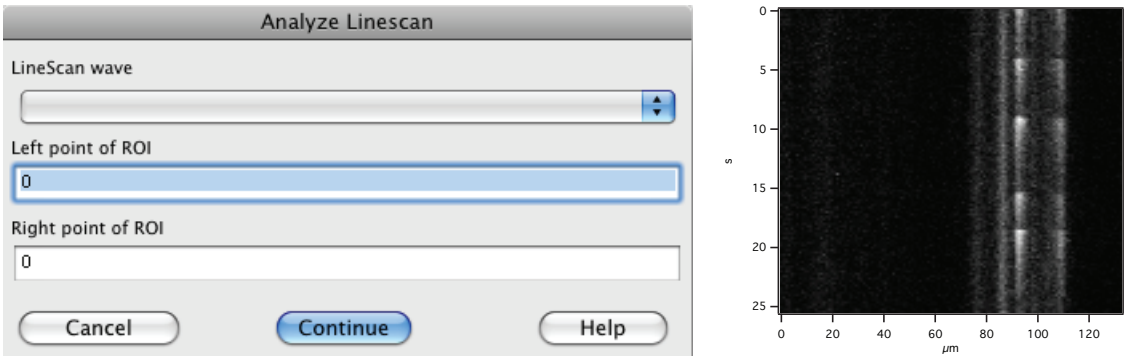

Fig. 12: The Linescan analysis control panel (left) and an example of linescan data (right).

12 Hierarchical clustering

Hierarchical clustering can be performed on a populationwave (see Section 5). Start by selecting **Hierarchical Clustering** from the SARFIA menu. This will open a panel (Fig. 13) to specify the data to analyse, the metric, whether data should be normalised and whether clusters that have only one member should be displayed. The following metrics can be chosen: Pearson, Euclidean, normalised Euclidean, Chebyshev, Hamming, Binned Data, and Manhattan. Pearson calculates the Pearson distance, which is defined as 1− Pearson's r. Thus, a distance of 0 always indicates perfect correlation, regardless of the metric used. Normalised Euclidean normalises by the standard deviation. Binning smoothes and bins the fluorescence data into 5 bins. Hamming is useful only for categorical data – it calculates whether points are similar and assigns either a distance of 1 or 0 to any point. The distance matrix contains the sum of all distances for each pair of traces.

Normalise Data (0/1)? selects whether data will be normalised before the operation. Generally, it is preferable to perform clustering on data normalised to a baseline, as this operation will normalise the values of any trace to range from 0 to 1.
Display clusters with a single trace (0/1)? specifies whether clusters that contain

only one trace will be displayed. For large data sets this has the possibility to open a large number of graphs and is thus generally not recommended.

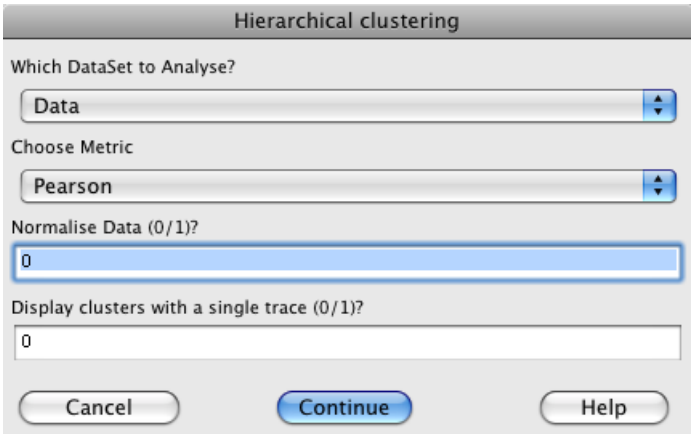

Fig. 13: The Hierarchical clustering control panel.

After the first control panel is dismissed, the distance matrix is displayed, together with a cumulative histogram of all distances in the data set (Fig. 14). Then, the user sets a threshold (the automatically generated value is 1/3 of the maximum distance). Traces with a distance less than the threshold to any other trace in a cluster are then merged with this cluster (single linkage or or nearest neighbour clustering). After dismissing the second control panel, the clusters are calculated. Results are stored in a separate data folder, termed "Clustering": Each cluster is saved as a populationwave containing all traces in that cluster, together with a wave for the average, standard deviation and standard error. The wave clusters stores the membership of each trace. Furthermore, the distance matrix, a modified populationwave, in which traces are sorted according to the membership to the clusters, the list of distances (DM2C) and the histogram (DM2C_s) are stored in this data folder. The waves DM2C and DM2C_S contain the distances in the first column, and the indices of the traces that they were calculated from in the second and the third column.

The nodes to construct a tree of clustered data can be calculated using the `HiClu2D (Sorted)` function, where Sorted is the wave DM2C_s. The result is stored as the wave Clusters2D.

13 Positioning

The **IPL Measurement Panel** can be accessed from the SARFIA menu. It is used to determine relative positions of regions of interest in a layered structure (i.e. the inner plexiform layer [IPL] of the retina). Start by thresholding an image to calculate a MultiROI mask (see Subsection 4.2). In the control panel, select the image and the ROI

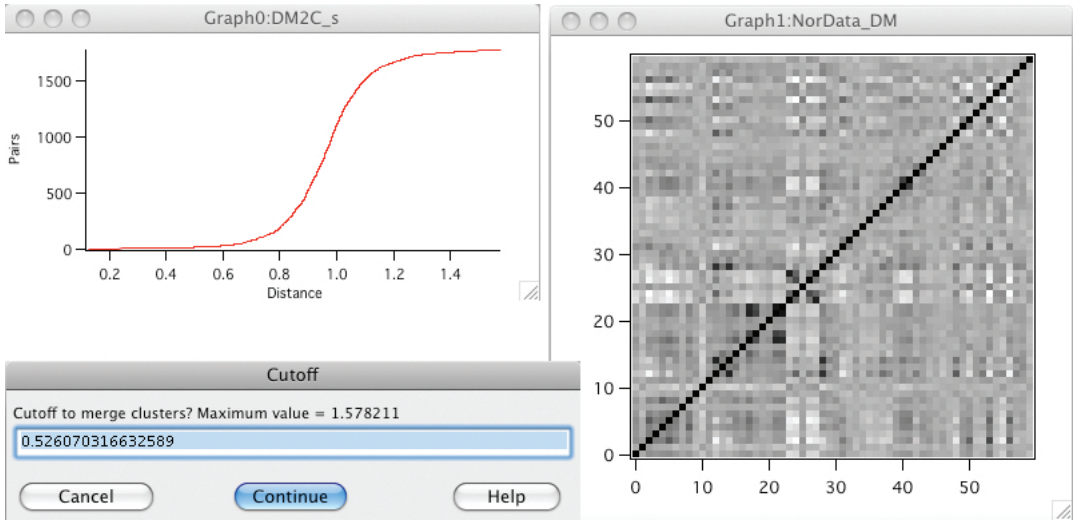

Fig. 14: Preview of the hierarchical clustering results: Cumulative histogram of the distances (top left), distance matrix (right) and setting of the threshold to merge clusters (bottom left).

mask from the appropriate popup menus and select a method to determine the borders of the layered structure. Manual drawing is generally recommended, since it is the only method that works reliably on any data. Click start and either draw the borders, one at a time, or adjust the threshold until satisfied with the preview.

Next, select which of the two borders (automatic methods mat display three "fragments") faces top (i.e. 100%) and which one bottom (i.e. 0%) and click continue. When successful, the results are displayed in the preview window. It is advisable to check the results more carefully, using the **Show Results** button: This will display some of the isocontours that were calculated in order to determine the positions. If they vary visibly in length, or are jagged, the process should be repeated, starting with re-drawing the borders. This may happen, if the borders are diagonal in the image, and the program determines one of them to be horizontal and the other one vertical. Alternatively, rotate the image and ROI mask using the Matrix Rotation method (Section 6) before calculating the positions.

14 Database

The database can be used to combine any number of populationwaves (see Section 5) regardless of the sampling frequency and number of points. Additionally, associated information can be stored. Currently, the following fields are available: ROINr, Age, Position, Size, ONOFF, TSus, Stim, Baseline, Experimenter, AnalysisBitMask, nPoints, XDelta, XOffset, XUnit and OriginID. These are either determined automatically when

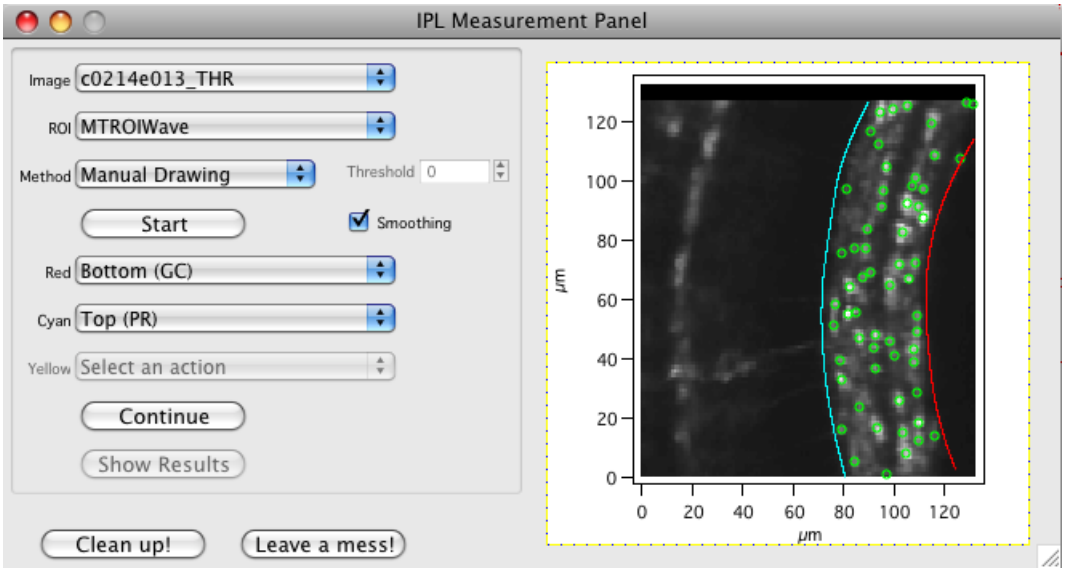

Fig. 15: Measurement panel to determine positions in a layered structure. The preview shows centres of mass (green circles) and manually drawn borders (cyan and red).

the database is generated, or they can be entered by the user (see Tab. 1).

The database is essentially a 2D wave that stores single traces in columns, like a populationwave. Information at the end of the database, the "footer", (see Fig. 16 is used to restore the original wave scaling and number of points. The popwavebrowser can display traces from a database and can be used to edit the associated information.

14.1 Adding data to a database

Data can be added to the database by choosing **Add to Database** from the SARFIA menu. This launches four consecutive control panels (Fig. 17) in which the appropriate waves and variables are specified. If the respective automatically named waves are in the active data folder, they will be automatically recognised.

Databases can be merged in two ways: Either an existing database is chosen from the **Basic Data** control panel (Fig. 17). Then, the new information will be appended to that database. Alternatively, the `CombineDataBases(ListWave)` command can be used. This needs a wave reference wave (`make/wave`) passed as a parameter. The following example describes how to combine the databases, DataBase1, DataBase2 and NewDataBase.

```
make/o/wave ListWave
ListWave = {DataBase1, DataBase2, NewDataBase}
```

Label	Function	Automatic?
`ROINr`	ROI number in the original context	×
`Age`	Age of the test subject	(×)
`Position`	ROI position	(×)
`Size`	ROI size	×
`ONOFF`	Response classification 1	–
`TSus`	Response classification 2	–
`Baseline`	F_0 (if data was normalised)	(×)
`Experimenter`	Name of experimenter from list	(×)
`AnalysisBitMask`	Analysis parameters, entered as a bit mask	(×)
`nPoints`	Number of data points in trace	×
`XDelta`	DimDelta of trace	×
`XOffset`	DimOffset of trace	×
`XUnit`	WaveUnits of trace	×
`OriginID`	Name of the populationwave[1]	×

Tab. 1: Associated information stored in a database. ×, information is automatically stored; (×), information can be entered when the database is generated; –, information must be supplied as waves when the database is generated; [1], capitalisation and characters other than the roman alphabet, numbers and the underscore character will be lost.

```
CombineDataBases(ListWave)
```

The resulting combined database will be called "w_BDWave".

14.2 Retrieving data from a database

The basic functions to retrieve data from a database are `TraceFromDB(DataBase,index,[ResultName])`, to retrieve a single trace, and `PopFromDB(DataBase,[ResultName])` to generate a populationwave from a database. The latter, however, is possible only if all entries in the database have the same number of points and scaling.

A third, and possibly the most useful, way to retrieve data is by writing a custom function that retrieves a subset of the data based on the associated information, for instance retrieving all data that were recorded at a particular age or age interval, and had a specific stimulus applied. A template of such a function is provided in the file `ExpDB2_Extraction.ipf`, which is automatically loaded with the SARFIA procedures.

R1106 Label		nPoints			
Row	DataBase.I	DataBase[][0].d	DataBase[][1].d	DataBase[][2].d	DataBase[][3].d
	x \ y				
1082		13.3153	3.01896	1.7329	4.26484
1083		10.8707	5.28337	3.5917	2.34308
1084		11.3918	4.22866	2.36418	2.5263
1085		12.7982	6.77926	0.458611	4.06462
1086		9.76747	6.35438	0.706542	3.71331
1087		11.3667	4.00362	2.70187	1.31875
1088		9.84021	4.686	3.37699	2.35433
1089		10.513	6.32277	3.88185	5.3464
1090		11.2226	6.61876	3.52236	3.15569
1091		9.83634	2.21831	2.13684	2.01462
1092		13.3153	5.50323	1.98576	3.21279
1093		10.8707	4.42542	3.89512	2.38579
1094		11.3918	3.3473	2.98164	1.68151
1095		13.5816	4.47321	1.7366	4.30983
1096	ROINr	0	1	2	3
1097	Age	9	9	9	9
1098	Position	28	25	78	55
1099	Size	8.6528e-11	8.6528e-11	8.6528e-11	8.6528e-11
1100	ONOFF	0	0	2	1
1101	TSus				
1102	Stim	6090	6090	6090	6090
1103	BaseLine	9.61887	4.82172	1.84239	2.2915
1104	Experimenter	6	6	6	6
1105	AnalysisBitMask				
1106	nPoints	1096	1096	1096	1096
1107	XDelta	0.2	0.2	0.2	0.2
1108	XOffSet	0	0	0	0
1109	XUnit	0	0	0	0
1110	OriginID	2.14804e+15	2.14804e+15	2.14804e+15	2.14804e+15
1111					

Fig. 16: Organisation of data in a database. The leftmost column shows the dimension labels.

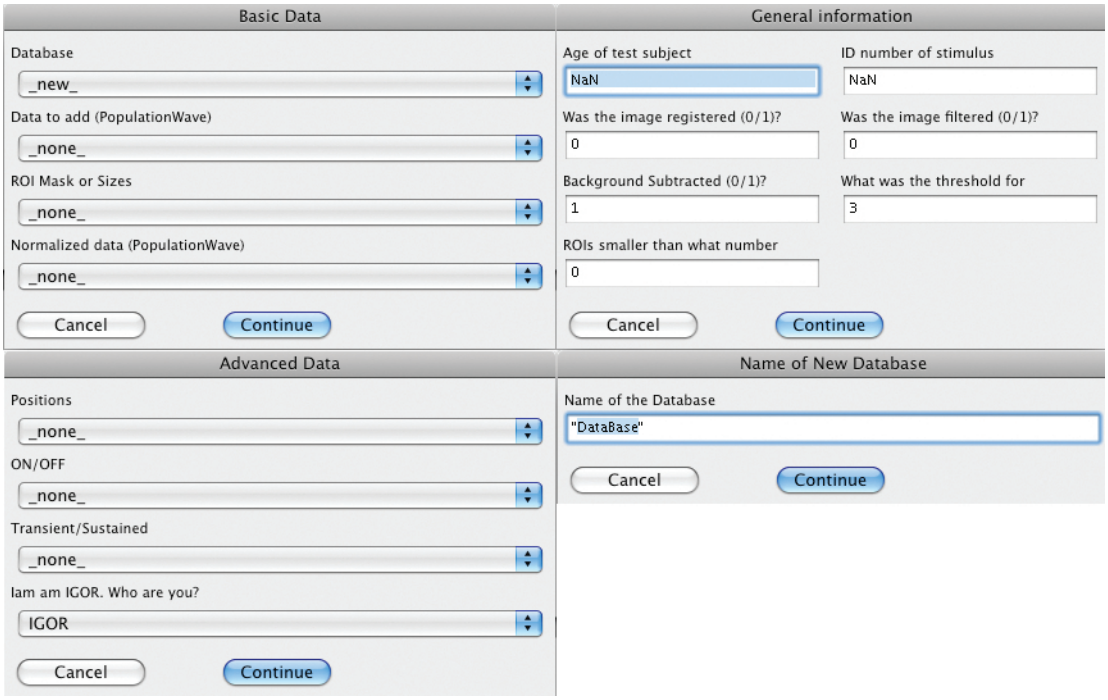

Fig. 17: Control panels to generate a database.

15 Known issues

- **IPL Measurement**: If the the structure is diagonal in the image, calculation of the isocontours may fail. Re-draw the borders or rotate the image and ROI mask (see also Section 13).
- **Matrix Rotation**: Sometimes the rotated image may have a line too few, probably due to a rounding error (see, for instance, Fig. 10).
- **Quick Analysis Notebook**: If quick analysis is done more than once on the same data set, the results displayed in the notebook may be from the previous run. Close all layout windows (and possibly all unused graph windows), using the Igor Graph Browser, before re-running a quick analysis.

Index