# MetaData

## An ImageJ Plugin

## A Brief Guide

Sean Parsons

McMaster University

version 1 August 2010

version 2 April 2011

# 1 INTRODUCTION

## 1.1 Rationale

Metadata is data associated with an image rather than coded by the image itself. This might include what algorithm created the image, dates and times and, most importantly for quantitative image analysis, the scaling of the image - pixel/intensity resolutions and origins. Standard ImageJ has limited incorporation of image metadata. I wrote the MetaData plugin specifically to do this.

## 1.2 Metaobjects, Metapictures, Metatables and Metadata

MetaData operates with two basic data types - metapictures and metatables - collectively called **metaobjects**.

A **metapicture** is a single image and its associated metadata. They are saved as tiff files with the metadata contained in the ImageDescription tag of the tiff. Metapictures can also be associated with regions of interest (roi's) but I have not yet implemented a way of saving these.

A **metatable** contains columns of data (each column with a title and units), a list of variables and its associated metadata. They are saved as tab-delimited txt files. The first two columns of the file contain the metadata, the next two columns the variables and the remaining columns the column data (with the first two rows containing the column titles and units).

A metaobject's **metadata** consists of several fields:

**Class ID**

*This will always be either "METAPICTURE" or "METATABLE".*

**Unique ID**

*A random string used to uniquely identify the metaobject.*

**Creation Process**

*A description of the algorithm or process used to create the image.*

**Data Type**

*A description of what the picture or table represents.*

**Creation Date**

*The date and time the metaobject was created.*

**Parent's ID**

*If the metaobject was created by an algorithm that had another metobject as input, this contains the Unique ID of that metaobject.*

For metapictures there are also:

**x name**

*The name of the x dimension. e.g. space.*

**x unit**

*The units of the x dimension. e.g. microns.*

**x origin**

*The origin of the x dimension in x units. e.g. 5.0 microns.*

**x resolution**

*The pixel resolution of the x dimension in x units per pixel. e.g. 2.5 microns/pixel.*

These fields are repeated for the "y" and "a" dimensions - the latter is the amplitude or intensity of the image.

Any number of further fields can be added by particular algorithms. Those listed above are just the standard minimum. If a tiff file is opened with MetaData and there is no associated metadata, default metadata is assigned.

# 2 INSTALLATION

Version 1.43 or higher of ImageJ must be installed.

Once ImageJ is installed, close it and copy the following library files to the "jre\lib\ext" folder of ImageJ:

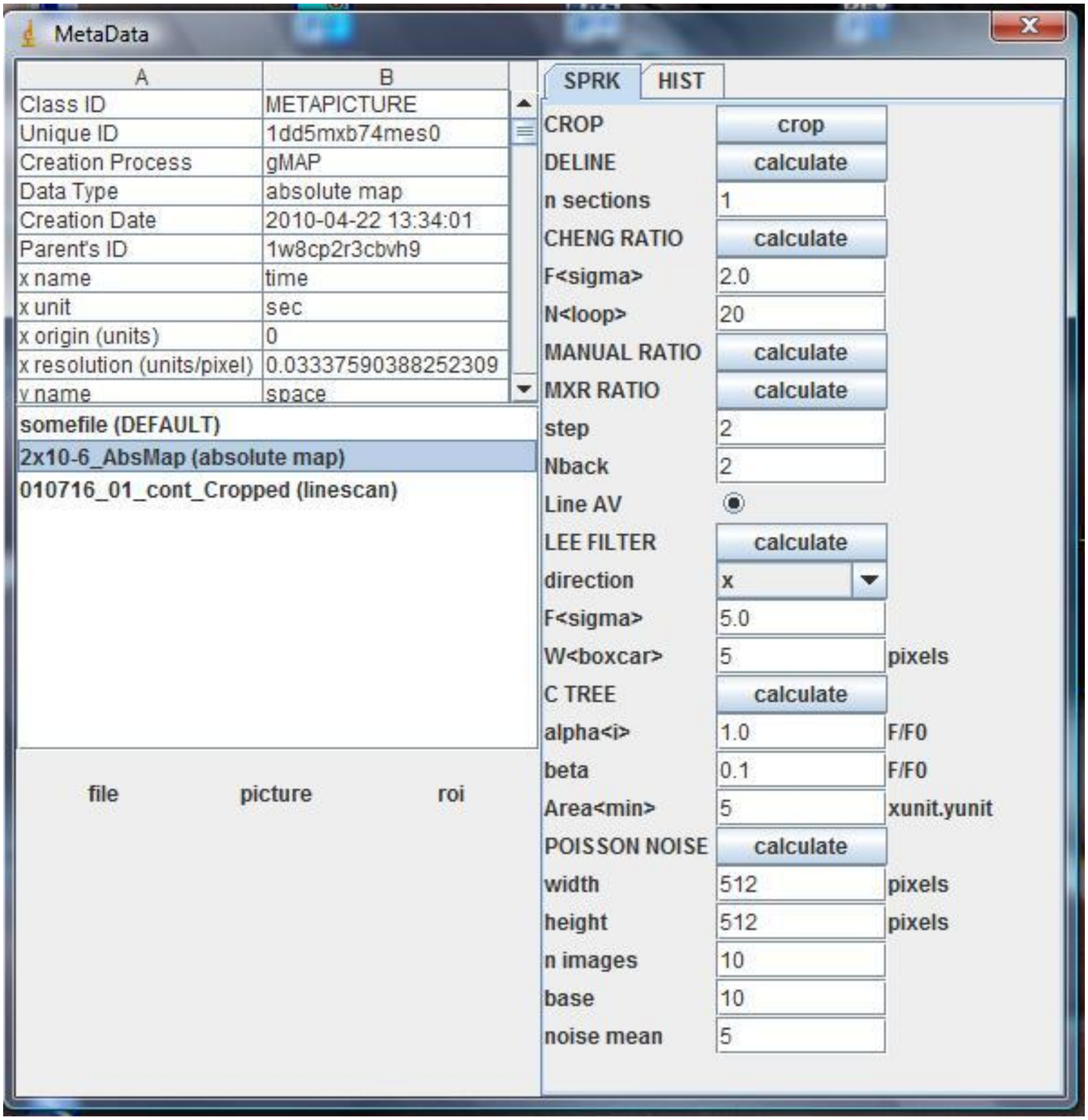
- flanagan.jar
- jtransforms.jar
- marfmath.jar

Copy LineScan_Analysis.jar to the "plugins" folder of ImageJ.

Open ImageJ. Go to Plugins>MetaData_V1 and the MetaData plugin opens.

# 3 PLUGIN INTERFACE

The MetaData plugin interface is divided into a left and right panels - the object panel and algorithm panel, respectively.



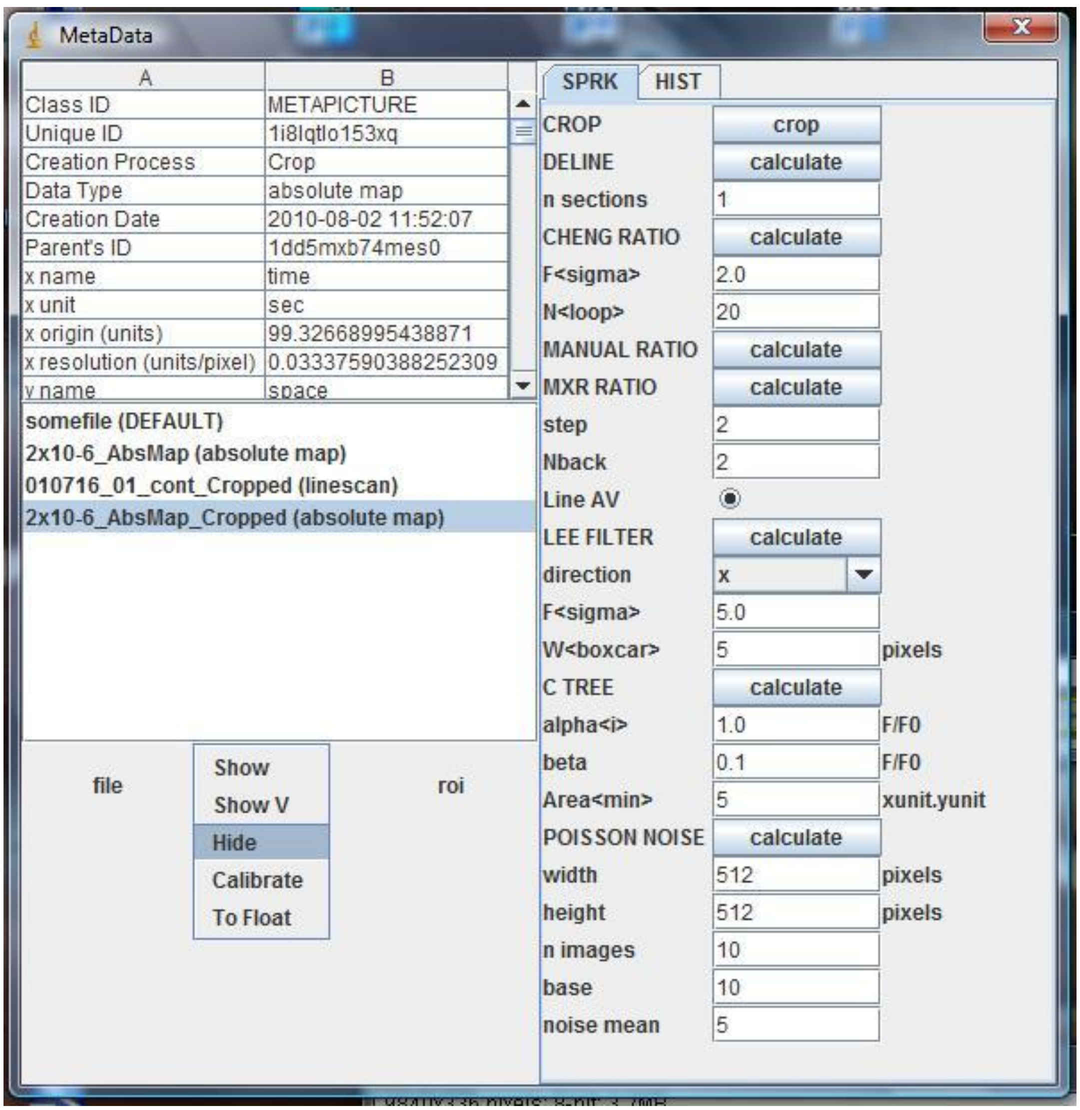
### 3.1 Object Panel

This is divided into three vertical panels.

The **middle** panel contains a list of those metaobjects currently open. Objects can be selected individually or in groups by holding ctrl or shift or shift-A.

The **top** panel displays the metadata associated with the metaobject selected in the middle panel. The user can change certain "unprotected" metadata fields by typing a value into the relevant field of the panel. e.g. x resolution, y origin, etc. (see also *Calibrate* command, below).

The **bottom** panel contains three words - file, picture and roi. By clicking any mouse button over one of these words a drop-down menu is opened. Each menu contains commands which act on the metaobject selected in the middle panel.



file.......

**Open**

*Opens a tiff or txt file. If metadata cannot be retrieved the object is assigned default metadata. If the structure of the file does not otherwise conform to a metaobject a default metaobject is created.*

*Note: some tiff's may not open due to format issues. If this occurs it often works to first open and resave the tiff through the ImageJ menu, then try and open through MetaData.*

**Close**

*Closes a metaobject. Does not automatically save any changes.*

**Save**

*Save a metaobject.*

**SaveAs**

*Save a metaobject with a particular path/name.*

**Copy**

*Copies a metaobject. A new Unique ID and Creation Date is assigned.*

picture......

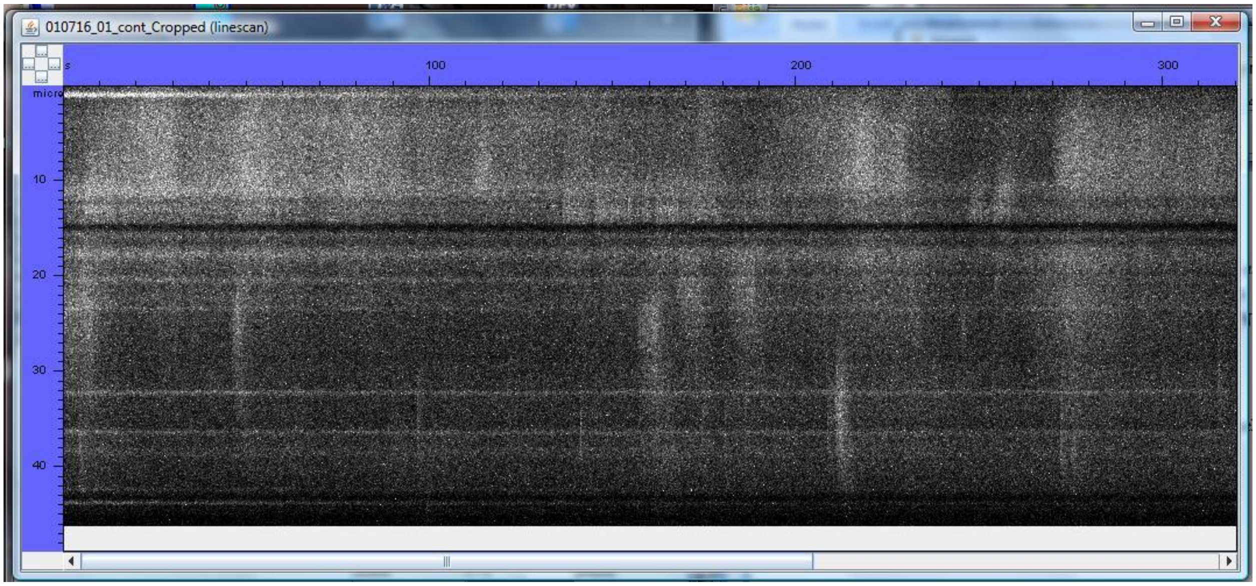
These commands are only responsive to metapictures selected in the middle panel.

**Show**

*Displays the metapicture's image in a standard ImageJ window. This window should only be closed by Hide. If it is closed by the window's X button or from ImageJ, pressing Show will not reopen it.*

**ShowV**

*Displays the metapicture's image in a scrollable window. The blue rulers along the left and top boundaries measure the image according to the dimension origins and resolutions contained in the metadata. The upper left corner contains buttons for zooming the image in the x and y dimensions separately. This will also cause resizing of the standard ImageJ window image if it is opened. This window is not closed by Hide.*



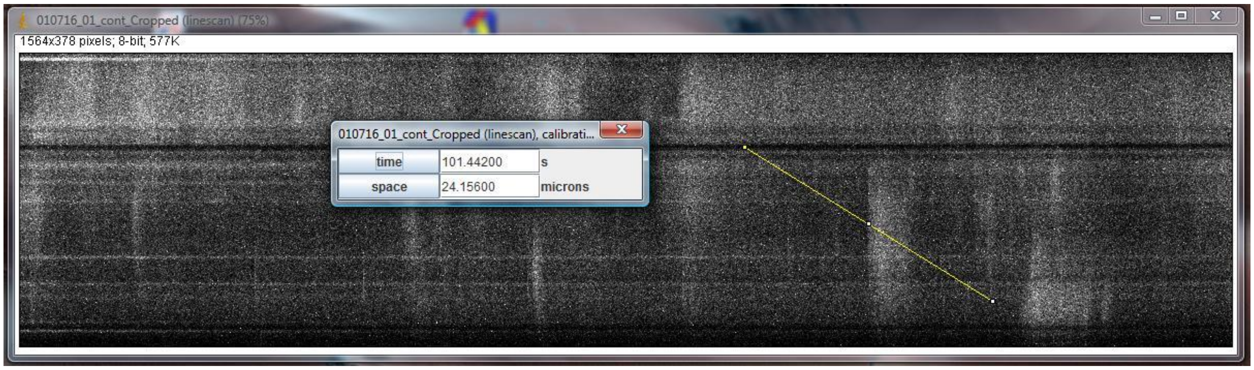

**Hide**

*Hides the standard ImageJ window displaying a metapicture's image.*

**Calibrate**

*Opens a dialog which allows the user to measure distances in the image and to adjust its x or y resolution. The dialog contains either one row (if the x and y dimensions have the same name and units) or two rows (if they are different). Each row contains a button labelled with the name of the dimension followed by a text input box followed by the unit name. By selecting one of the ImageJ roi tools, dragging the roi across the image updates the text box(s) with the size of the roi.*

*If the user types a new value in the text box and then presses the button labelled with the dimension name, a dialog will appear asking the user to confirm that they wish to change the dimension's resolution. Pressing "Okay" changes the resolution according to the typed value and the pixel size of the roi.*



**To Float**

*Converts the image to a 32bit Floating-point representation. "a" resolution and origin are set to 1 and 0,  respectively.*

roi......

These commands allow the user to copy and paste roi's between metapictures.

**To Clipboard**

*Copies the selected metapictures roi's to a clipboard.*

**From Clipboard**

*Pastes the clipboard's roi's to the selected metapicture(s).*

**Clear Clipboard**

*Clears the clipboard.*

**Clear Current**

*Clears all roi's from the selected metapicture(s).*

## 3.2 Modifying and Saving Images

Images can be modified a number of ways - by zooming with the ShowV window and, if the image is displayed in an ImageJ window (Show command), by any ImageJ command or plugin. If the image is displayed in an ImageJ window it can also be saved from the ImageJ menu.

However there are some important points here:

1) When saving with the MetaData Save or SaveAs commands only the unmodified, original image is saved. This is to protect the original data.

2) When saving with the ImageJ menu, no metadata is saved, and the image is saved as displayed (i.e. with any modifications).

### 3.3 Algorithm Panel

This consists of a set of tabbed panes, each containing inputs for one or more algorithms. When an algorithm has finished, its output of metaobject(s) are added to the list in the Object Panel.

The layout of each panel and operation of each algorithm is peculiar to the algorithm, but the general rules are:

1) Text inputs and radio-buttons set the algorithm parameters.

2) A button labelled "calculate" initiates an algorithm.

3) When the algorithm is working the button label changes to "processing...." and is unresponsive until the algorithm finishes at which point the label reverts to "calculate".

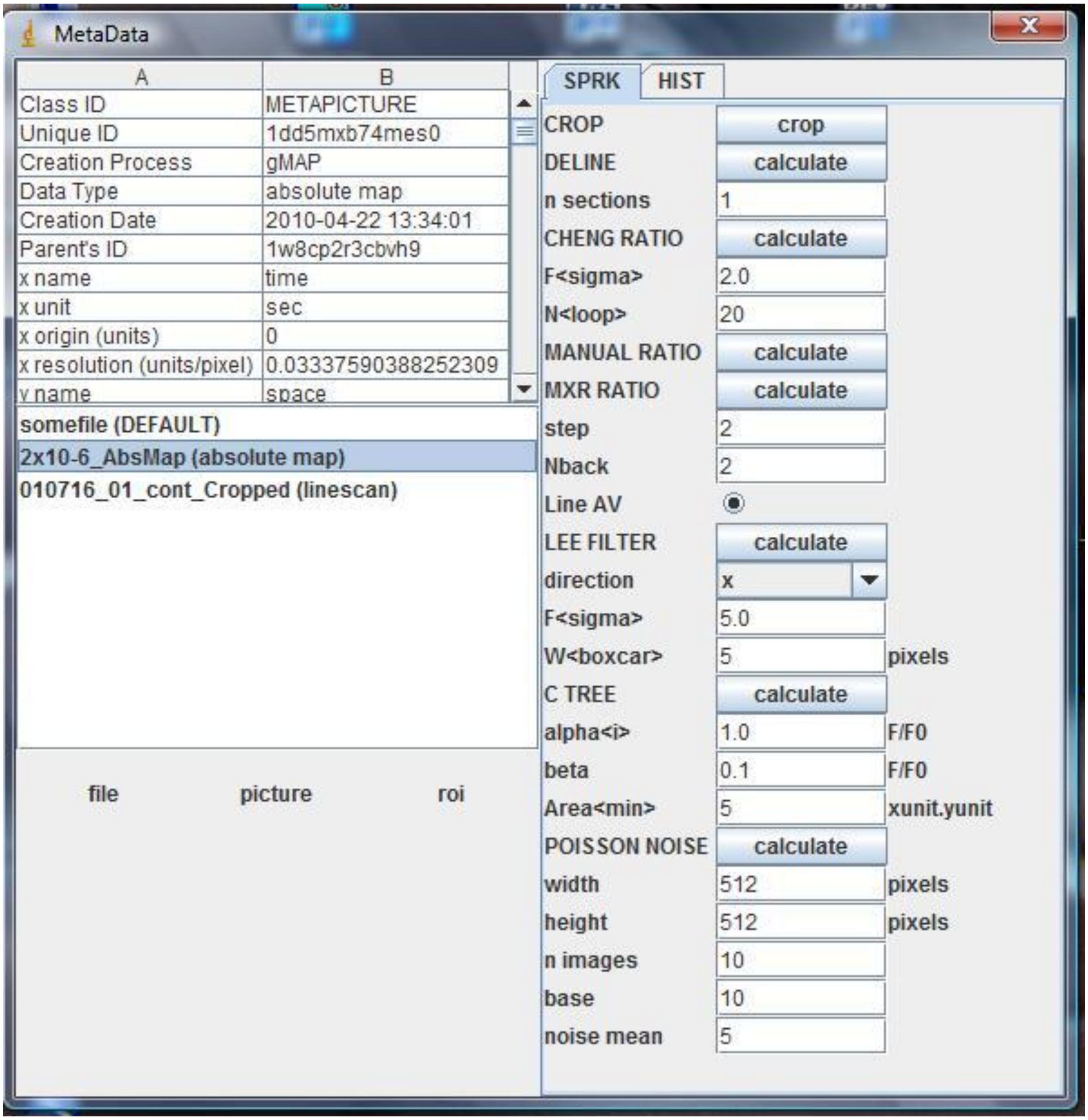
4) Algorithms never modify the input data - they always output new data.

5) Some algorithms will work in a loop on all the metaobjects selected in the Object panel. Others will just work on the first selected object.

6) Output metapictures are given the name of the input picture concatenated with a brief description of the algorithm, _xxxx, where xxxx is the description. Thus a series of three algorithms might give a picture named "picture_algo1_algo2_algo3". This provides a simple way of indicating the algorithmic history of an image.

7) Any ROI annotation of an output metapicture, usually only appears after resizing the image with Show V. After this it will remain visible however the metapicture is resized.

# 4 SPARK ALGORITHM PANEL



From top to bottom the algorithms are:

**Crop**

Crops according to an roi selected from ImageJ tools.

**Deline**

Bands of brightness along the time dimension are common in line scans of calcium fluorophore fluorescence, reflecting loading of organelles. Izu *et al.* (*Biophys J* 75: 1144-1162, 1998) removed these bands by zeroing the zero-frequency component of each time line's Fourier transform. This algorithm is a modification of this.

For each row of pixels, the FFT is calculated for **n sections** of equal length along that row. The zero-frequency components of each section are linearly interpolated and the function of this line is subtracted from the untransformed row.

**Cheng Ratio**

This implements the normalization algorithm of Cheng *et al.* (*Biophys J* 76: 606-617, 1999). In Cheng et al. the mask excises pixels > mean + 2$\sigma$, where $\sigma$ is the standard deviation. In our algorithm the mask excises pixels > mean + **F<sigma>**$\sigma$. **N<loop>** is the maximum number of iterations of the normalization loop.

**Manual Ratio**

The image is normalized by $F_0$ where $F_0$ is the average pixel value of the roi selected by the user.

**MXR Ratio**

Each row of pixels segmented at different thresholds (T), beginning at the minimum pixel value and then increasing in steps of **step**. At each threshold the number of segmented regions is counted. If the number of regions at a threshold ($T_{current}$) is less than the number counted at T = $T_{current}$ - **Nback**<**step**>, the algorithm stops and $F_0 = T_{current}$ - **Nback**<**step**>. If **LineAV** is not selected, each row is divided through by its own $F_0$. If **LineAV** is selected the whole image is divided through by the average of these $F_0$ values.

**Lee Filter**

Implements the filter of Lee (*Opt Eng* 25: 636-643, 1986). **F<sigma>** is the s factor and **W<boxcar>** is the pixel width of the kernal. The **direction** drop-down list selects whether the kernal is one dimensional (either x or y) or two dimensional (xy).

**C Tree**

Implements the confinement tree algorithm of Parsons and Bolton (*J Physiol* 554: 687-705, 2004) for measurement of sparks. **Alpha<i>** = $\alpha_i$, **beta** = $\beta$, **Area<min>** = XTR. The output consists of a metatable containing metric data and a metapicture which is a copy of the input image with red rois marking event boundaries at half-maximum amplitude and green rois marking fwhm and fdhm. For single-pixel events boundaries are not marked.

<u>ROIS WILL ONLY APPEAR AFTER RESIZING THE METAPICTURE WITH SHOW V.</u>

The metatable containing metrics can be opened by dragging into Excel or Origin.

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | METADATA | | VARIABLES | |
| 2 | Class ID | METATABLE | alpha<i> (arb) | 1 |
| 3 | Unique ID | 7jt81yescfov | beta (arb) | 0.01 |
| 4 | Creation Proc | CTree | min area (pixels) | 3 |
| 5 | Data Type | metric data | min area (microns.: | 0.1 |
| 6 | Creation Date | ############## | | |
| 7 | Parent's ID | 1qnpes6jbcevp | | |
| 8 | | | | |
| 9 | | | | |
| 10 | | | | |

| | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | DATA | | | | | | | | | | | | | | | | | | | | |
| 2 | max(t) | max(x) | b(t1) | b(t2) | b(x1) | b(x2) | MA | MA-1 | hMA | AHM | MHM | FDHM | FWHM | log(MA) | log(MA-1) | log(hMA) | log(MHM/ | log(AHM) | log(MHM) | log(FDHM | log(FWHM) |
| 3 | s | microns | s | s | microns | microns | arb | arb | arb | microns.s | microns.s. | s | microns | arb | arb | arb | arb | microns.s | microns.s. | s | microns |
| 4 | 1174.692 | 4.636 | 1174.692 | 1175.01 | 4.636 | 4.758 | 1.010697 | 0.010697 | 1.005348 | 0.116388 | 6.22E-04 | 0.318 | 0.122 | 0.004621 | -1.97074 | 0.002317 | -2.27178 | -0.93409 | -3.20587 | -0.49757 | -0.91364 |
| 5 | 324.678 | 9.394 | 324.678 | 324.996 | 9.394 | 9.516 | 1.010697 | 0.010697 | 1.005348 | 0.116388 | 6.22E-04 | 0.318 | 0.122 | 0.004621 | -1.97074 | 0.002317 | -2.27178 | -0.93409 | -3.20587 | -0.49757 | -0.91364 |
| 6 | 258.216 | 25.62 | 257.898 | 258.534 | 25.498 | 25.742 | 1.010697 | 0.010697 | 1.005348 | 0.116388 | 6.22E-04 | 0.636 | 0.244 | 0.004621 | -1.97074 | 0.002317 | -2.27178 | -0.93409 | -3.20587 | -0.19654 | -0.61261 |
| 7 | 1305.072 | 34.77 | 1305.072 | 1305.39 | 34.77 | 34.892 | 1.010697 | 0.010697 | 1.005348 | 0.116388 | 6.22E-04 | 0.318 | 0.122 | 0.004621 | -1.97074 | 0.002317 | -2.27178 | -0.93409 | -3.20587 | -0.49757 | -0.91364 |
| 8 | 1017.282 | 35.746 | 1016.964 | 1017.6 | 35.624 | 35.868 | 1.010697 | 0.010697 | 1.005348 | 0.116388 | 6.22E-04 | 0.636 | 0.244 | 0.004621 | -1.97074 | 0.002317 | -2.27178 | -0.93409 | -3.20587 | -0.19654 | -0.61261 |
| 9 | 1140.666 | 36.966 | 1140.666 | 1140.984 | 36.966 | 37.088 | 1.010697 | 0.010697 | 1.005348 | 0.116388 | 6.22E-04 | 0.318 | 0.122 | 0.004621 | -1.97074 | 0.002317 | -2.27178 | -0.93409 | -3.20587 | -0.49757 | -0.91364 |
| 10 | 49.608 | 54.778 | 49.608 | 49.926 | 54.778 | 54.9 | 1.010697 | 0.010697 | 1.005348 | 0.116388 | 6.22E-04 | 0.318 | 0.122 | 0.004621 | -1.97074 | 0.002317 | -2.27178 | -0.93409 | -3.20587 | -0.49757 | -0.91364 |
| 11 | 1369.944 | 7.198 | 1369.944 | 1370.262 | 7.198 | 7.32 | 1.036284 | 0.036284 | 1.018142 | 0.116388 | 0.002112 | 0.318 | 0.122 | 0.015479 | -1.44029 | 0.007808 | -1.74132 | -0.93409 | -2.67541 | -0.49757 | -0.91364 |
| 12 | 318.318 | 7.442 | 318 | 318.636 | 7.32 | 7.564 | 1.036284 | 0.036284 | 1.018142 | 0.116388 | 0.001615 | 0.636 | 0.244 | 0.015479 | -1.44029 | 0.007808 | -1.85769 | -0.93409 | -2.79178 | -0.19654 | -0.61261 |
| 13 | 1156.884 | 7.808 | 1156.884 | 1157.202 | 7.808 | 7.93 | 1.036284 | 0.036284 | 1.018142 | 0.116388 | 0.001119 | 0.318 | 0.122 | 0.015479 | -1.44029 | 0.007808 | -2.01715 | -0.93409 | -2.95124 | -0.49757 | -0.91364 |
| 14 | 116.388 | 37.088 | 116.388 | 116.706 | 37.088 | 37.21 | 1.036284 | 0.036284 | 1.018142 | 0.116388 | 0.001615 | 0.318 | 0.122 | 0.015479 | -1.44029 | 0.007808 | -1.85769 | -0.93409 | -2.79178 | -0.49757 | -0.91364 |
| 15 | 551.094 | 37.21 | 550.776 | 551.412 | 37.088 | 37.332 | 1.036284 | 0.036284 | 1.018142 | 0.116388 | 0.001615 | 0.636 | 0.244 | 0.015479 | -1.44029 | 0.007808 | -1.85769 | -0.93409 | -2.79178 | -0.19654 | -0.61261 |

The data columns are,

max(x), max(t) - coordinates of the event peak.

b(t1), b(t2), b(x1), b(x2) - coordinates of the event's half-maximum boundary at points coincident with the peak (i.e. the start and ends of the fwhm and fdhm).

MA, MA-1, hMA - peak amplitude, peak amplitude minus 1, half peak amplitude.

AHM, MHM - area at and mass above half peak amplitude.

FDHM, FWHM  - full duration and width at half peak amplitude.

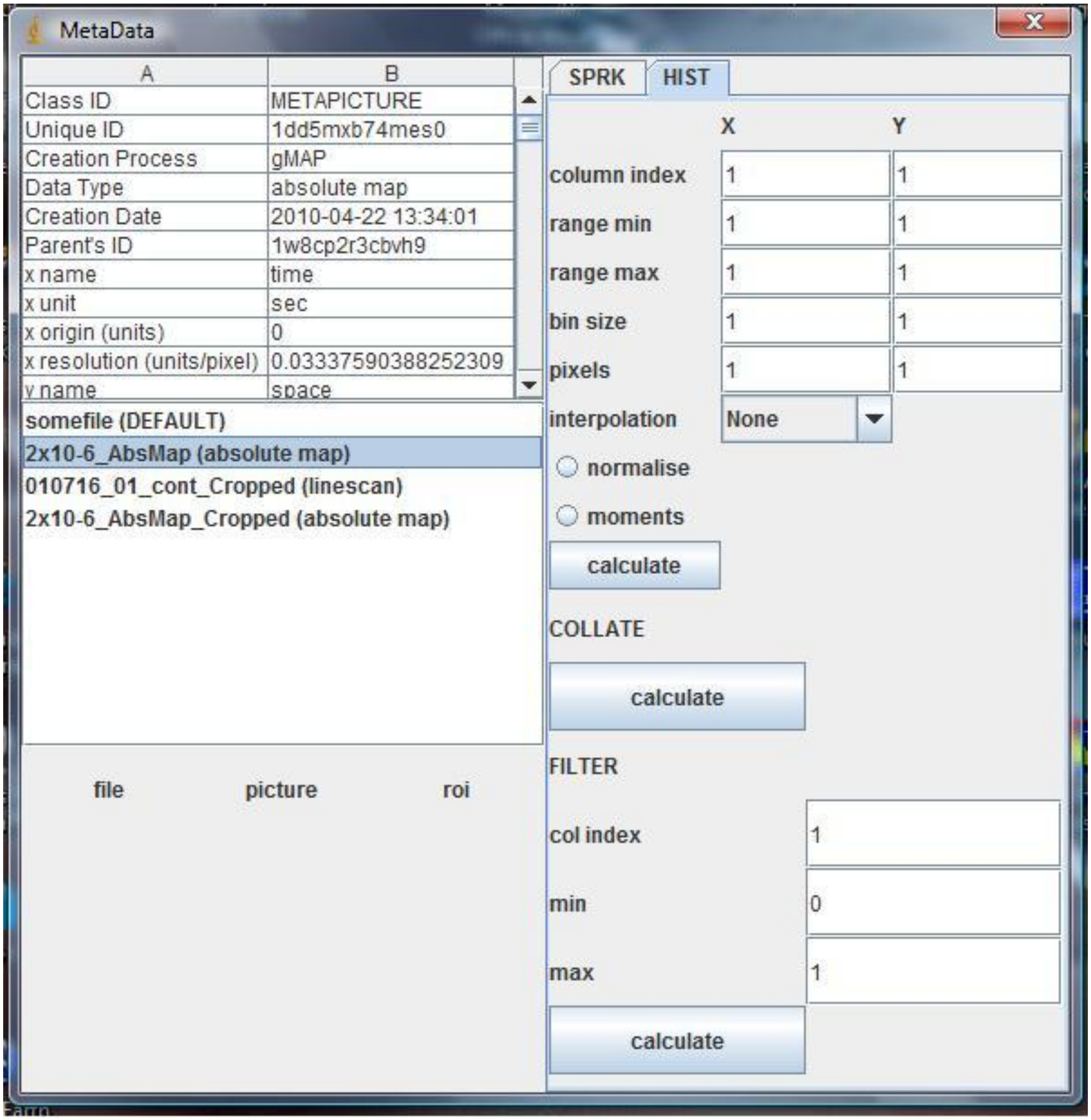
...*subsequent columns* - log of previous columns.

## Poisson Noise

Generates *de novo* images with Poisson noise. **n images** are generated of **width** x **height** dimensions with Poisson noise generated by functions of Michael Thomas Flanagans Scientific Library with a mean of **noise mean**. **base** is then added to each image.

# 5 HISTOGRAM ALGORITHM PANEL

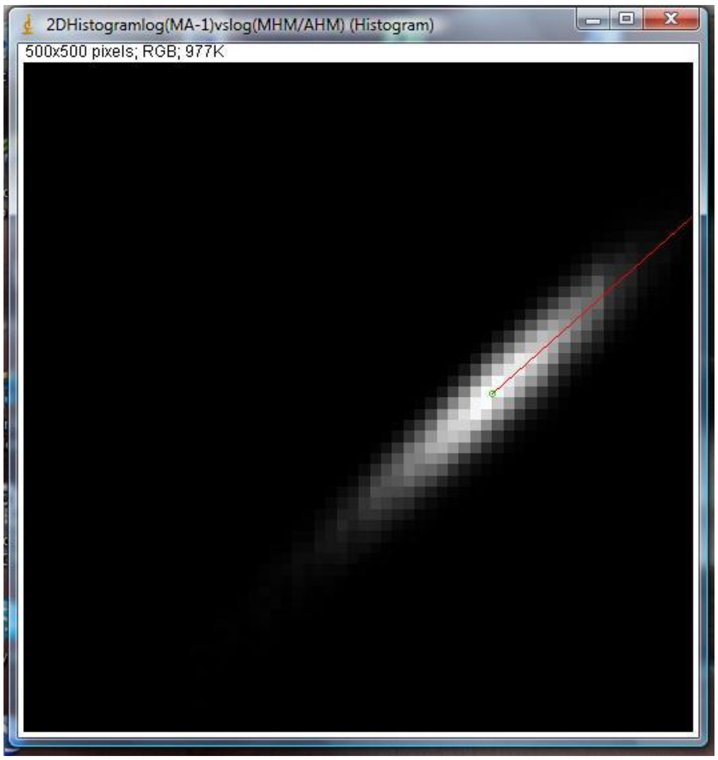
These algorithms operate on any metatable.



**Histogram**

This algorithm (top half of the panel) creates a two-dimensional histogram image of paired data contained in two columns of the input metatable. **column index** are the indexes of these columns (where the first data column = 1); **range min** and **range max** are the histogram minimum and maximum; **bin size** is the bin size; **pixels** is the final dimension of the output image; the **interpolation** drop-down allows for selection of interpolation between bins; selecting the **normalise** radio-button causes the histogram to be normalised (a probability distribution); selection of the **moments** radio-button causes the principle component analysis (PCA) of the distribution to be output as a metatable. The x-y centroid is marked as a green circle on the histogram and the major axis is marked as a red line from this centroid. If several input metatables are selected there is only one output metatable of moments with a single row for each input - the top row ("TOTAL") contains the PCA of the summed data.

<u>ANNOTATION  WILL ONLY APPEAR AFTER RESIZING THE HISTOGRAM WITH SHOW V.</u>



| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Formula Bar | 1 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | METADATA | | VARIABLES | | DATA | | | | | | | | |
| 2 | Class ID | METATABLE | | | file | N | centroid x | centroid y | minor axis | minor axis | major axis | major axis | eccentricity |
| 3 | Unique ID | 15iwyltdd5jgy | | | | | arb | arb | | arb.arb-1 | | arb.arb-1 | unitless |
| 4 | Creation F | Histogram2D | | | TOTAL | 224671 | 0.102112 | -0.48611 | 0.00582 | 0.88282 | 0.183521 | 0.882827 | 0.984015 |
| 5 | Data Type | Moments | | | Collated_Table_fundus | 224671 | 0.102112 | -0.48611 | 0.00582 | 0.88282 | 0.183521 | 0.882827 | 0.984015 |
| 6 | Creation D | ######## | | | | | | | | | | | |
| 7 | Parent's II | mdazpbwyzzqi | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | |
| 9 | | | | | | | | | | | | | |

For a data set consisting of paired values, $\{x_0, y_0\}$, $\{x_1, y_1\}$,......$\{x_n, y_n\}$,

$$M_{pq} = \sum_i^n x_i^p y_i^q$$

$$x_c = M_{10} / M_{00} \qquad y_c = M_{01} / M_{00}$$

$$\mu_{pq} = \sum_i^n (x_i - x_c)^p (y_i - y_c)^q$$

$$\mu'_{pq} = \mu_{pq} / \mu_{00}$$

$$\mathrm{cov} = \begin{bmatrix} \mu'_{20} & \mu'_{11} \\ \mu'_{11} & \mu'_{02} \end{bmatrix}$$

$$s_{major} = -\mu'_{11} / (\lambda_{major} - \mu'_{20})$$

$$s_{\min or} = -\mu'_{11} / (\lambda_{\min or} - \mu'_{20})$$

$$c_{major} = y_c - (x_c s_{major})$$

$$\varphi = \left| [1 - (\lambda_{\max} / \lambda_{\min})]^{0.5} \right|$$

where $M_{pq}$ is the moment, $x_c$ and $y_c$ are the x and y centroids (**centroid x** and **centroid y**), $\mu_{pq}$ is the central moment, $\mu_{pq}{}'$ is the normalized central moment, *cov* is the covariance matrix, $\lambda_{minor}$ and $\lambda_{major}$ are the larger and smaller eigenvalues of the covariance matrix (**minor axis eigenvalue** and **major axis eigenvalue**), $s_{major}$ and $s_{minor}$ are the slopes of the major and minor axis' (**major axis slope**), $c_{major}$ is the y-intercept of the major axis (**major axis intercept**) and $\varphi$ is the **eccentricity**.

### Collate

This concatenates all the selected metatables.

### Filter

This removes all rows from the input metatable except where the column of index **col index** (where the first data column = 1) has the value between **min** and **max**.

# 6 ACKNOWLEDGEMENTS

The MetaData plugin contains code and libraries from several third-party sources:

***Michael Thomas Flanagan's  Java Scientific Library***

http://www.ee.ucl.ac.uk/~mflanaga/java/index.html

This super library (flanagan.jar) was used for used for matrix algebra, complex numbers, random numbers and interpolation.

***Piotr Wendykier's JTransforms***

http://sites.google.com/site/piotrwendykier/software/jtransforms

This is probably the best and fastest Java FFT library out there! (jtransforms.jar)

***MARF Math***

http://marf.sourceforge.net/

This library (marfmath.jar) was developed  by students at Concordia University for audio-recognition work. I used its Hamming Window calculation function.

***Vladimir Vassilevsky's filter design library***

http://www.abvolt.com/

This is a C++ library for design of FFT filters, which I translated to Java. I don't have the exact website for downloading it anymore.

***Burger and Burge***

http://www.imagingbook.com/index.php?id=98

These two author one of the best Java-orientated Image Processing books out there. What's even better is that you can down load their source code for free! I used their lightening quick segmentation and contour code. Also used their Quicktime i/o library (http://staff.fh-hagenberg.at/burger/imagej/MovieIO.html).

***ImageJ***

http://www.rsbweb.nih.gov/ij/

Of course none of this would be possible without the largesse of the NIH and Wayne Rasband.

# 7 SOURCE CODE

All source code was written and compiled with NetBeans 6.8 (www.netbeans.org). Source code is divided into eight folders:

| | |
|---|---|
| **frequency** | FFT algorithms. |
| **histogram** | Histogram panel and algorithms. |
| **metapicture** | Metaobject and metadata classes, object panel interface, calibration dialog, etc. |
| **PhysiolPlot** | ShowV image viewer. |
| **spark** | Spark panel and algorithms. |
| **SPUtils** | Utility functions. |
| **vvfilter** | Java translation of Vladimir Vassilevsky's C++ filter design library. |
| **ch11** | Burger and Burge code for segmentation and contouring. |

MetaData_V1.java is the main plugin run class.

# 8 CONTACTS

If you have any queries I can be contacted at

parsons_llabh@yahoo.co.uk