LA-UR-16-25116
Approved for public release; distribution is unlimited.

Title: MAMA User Guide v2.0.1

Author(s): Gaschen, Brian Keith
Bloch, Jeffrey Joseph
Porter, Reid
Ruggiero, Christy E.
Oyen, Diane Adele
Schaffer, Kevin M.

Intended for: Report

Issued: 2016-07-15

# MAMA User Guide v 2.0.1

## Table of Contents

# PREFACE

Morphological signatures of bulk SNM materials have significant promise, but these potential signatures are not fully utilized. This document describes software tools, collectively called the MAMA (Morphological Analysis for Material Attribution) software that can help provide robust and accurate quantification of morphological features in bulk material microscopy images (Optical, SEM). Although many of the specific tools are not unique to Mama, the software package has been designed specifically for nuclear material morphological analysis, and is at a point where it can be easily adapted (by Los Alamos or by collaborators) in response to new, different, or changing forensics needs. There are number of requirements for this further development:

1. The software must be developed with the help of the user community. This will ensure that the analysis needs of the community are met, that there is community consensus in how we describe and quantify features in images, and that software is easy and intuitive to use by actual end users.
2. The software must produce consistent descriptive and quantitative analysis so that the tools provide the foundations for a distributed 'knowledge base', where image quantification method and results can be documented, validated and shared between nuclear forensic laboratories.
3. The software must be 'open' so that they exact mathematics behind all calculations can be understood, uncertainties can be determined, and analysis results can be validated to meet users required standards.

Additionally, we have identified a number of goals that we hope the software can achieve:

1. We hope to make image analysis easier for users, so that achieving consistent results requires less time, and can be done by non-experts. This will require incorporating new advances in user-driven data analysis.
2. We hope to be able to provide consistent results on any type of material image (particles, bulk solids, metals) at any scale, so that any images, even historic images in archives, can be added to the communities growing material databases.
3. We hope to develop additional visualization and analysis tools that combine image quantification results across multiple users, and multiple samples, and thereby help the user community identify statistically significant morphological signatures of process history.
4. We hope to be able to provide consistent, and easily usable, tools and methods for quantifying texture for materials that do not have independent particles.
5. We hope to be able to support community development of expanded analytical needs as our imaging technologies improve or analysis needs expand. For example, 3D image quantification will be needed as microscale 3D imaging technologies become more readily used.

The current release of the MAMA software only includes the image quantification, descriptions, and annotation functionality. Only limited information on a sample, its pedigree, and its chemistry are recorded inside this part of the software. This was decision based on initial feedback and the fact that there are several analytical chemistry databases being developed within the community. Currently MAMA is a standalone program that can export quantification results in a basic text

format that can be imported into other programs such as Excel and Access. There is also a basic report generating feature that produces HTML formatted pages of the same information. We will be working with collaborators to provide better integration of MAMA into their particular systems, databases and workflows.

***This software is a work in progress. We appreciate people's patience working with software that may sometimes not work as seamlessly as commercial software. We welcome any feedback and any level of community involvement (from users to software developers) in its development.***

# 1.0 CORE FEATURES

The MAMA software is designed to help a user quantify and describe particles and textures in an image of a material. There are different modes for using the software, depending on a user's desired analysis goal. The basic use structure is designed to allow 1) images to be uploaded, 2) the particulates or other regions (e.g. grains) to be separated from the background or each other (segmentation), and 3) then quantification and/or labeling applied to all or subsets of the 'segmented' part of the image.

The software is not designed to do all this automatically, without user intervention. Quantitative image analysis, particularly for a field like nuclear forensics, is not standardized (i.e. there are no image 'standards' to which you must conform), and the features of interest in the image can vary widely. This means quantitative image analysis must be done with a 'user-in-the-loop'.

The software automates as much as possible, but critical decisions, input, and corrections must be made by a person. This is somewhat akin to manually marked extended features in fingerprint analysis—a far more consistent image data set. In terms of software use, this means that there are places for user input to the image processing—such as sliders to adjust the amount of segmentation desired—that you must use. The software is not designed to 'get it right' in the first result presented. Sometimes, in simple images, it does, and no user refinement is needed. In most cases however, the software presents a results (such as an image segmentation) in the middle of the range of segmentations that are possible, with no analysis to suggest this is the 'right' segmentation. The user must then adjust the result based on their expertise and desired analysis result (i.e. is the user looking to segment out small sub-particles from a larger sintered particle, or attempting to segment out the larger particle as a single entity?). After this initial interactions, there may still be corrections needed (i.e. a segmentation missed, or a segmentation where there should not be one). Tools are included for doing these types of corrections. The amount of user-corrections will depend on the images and the analysis the user is trying to perform. This will be discussed more in with respect to segmentation in Section 1.6.

**A final important note on the software:** We have created the software to be extremely flexible at this stage, both in how analyses can be performed and in the software interface. Some of the user-interaction is a result of designing the software to be very flexible for the broad types of images and materials that the nuclear forensics may encounter. The software does not limit what types of images are used (there are as yet no quality standards or metrics, resolution requirements,

formatting requirements, number of particle requirements, instrumentation requirements, etc. for any type of analysis or customer.), nor make user follow set proscribed analysis pathways. This was a design choice, reflecting the need to be able to utilize large existing libraries of images, a broad range of analysis needed dependent on a material's attributes, and the need to capture expert knowledge in an unlimited image/morphology feature space. There are many places where hard coded limits could have been put into place to prevent possible user errors or software issues (e.g. forcing some windows to close), and many places that we did not put in complete "are you sure you want to do that?" user checks (e.g. overwrite a segmentation, quantify texture on a filtered image).

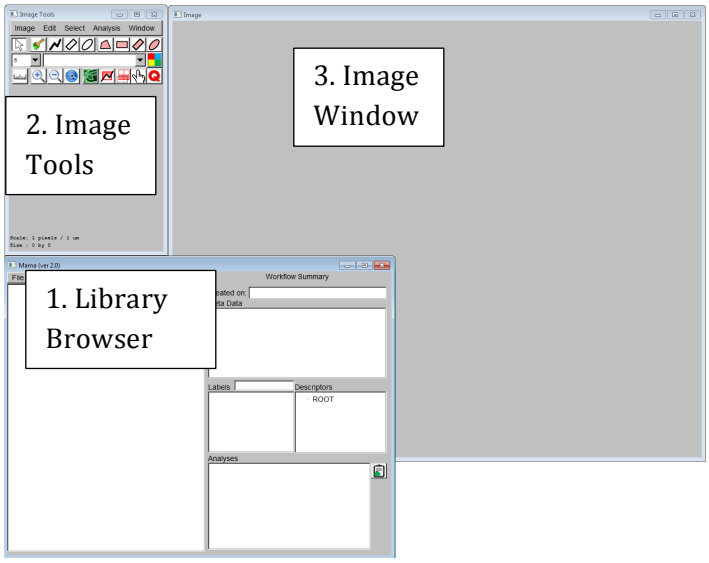
We expect that as particular use cases and analysis needs solidify from the community, the software will be modified to provide a more constrained set of tools targeting specific features, and this will enable us to provide a more robust tool in terms of software (less bugs) and in terms of user actions (users will not be able to get in trouble as much!).

## 1.1 Starting Mama and Interface Overview

Typically, mama is executed from a shortcut on your desktop or by selecting the program from the **Start Menu**:

**Programs -> Mama -> Mama.exe**

You can also create a desktop shortcut by simply dragging the Mama red 'M' icon from the start memo onto your desktop. Mama will launch with three GUI windows opened: 1) Library Browser, 2) Image Tools and 3) Image Viewer Window.



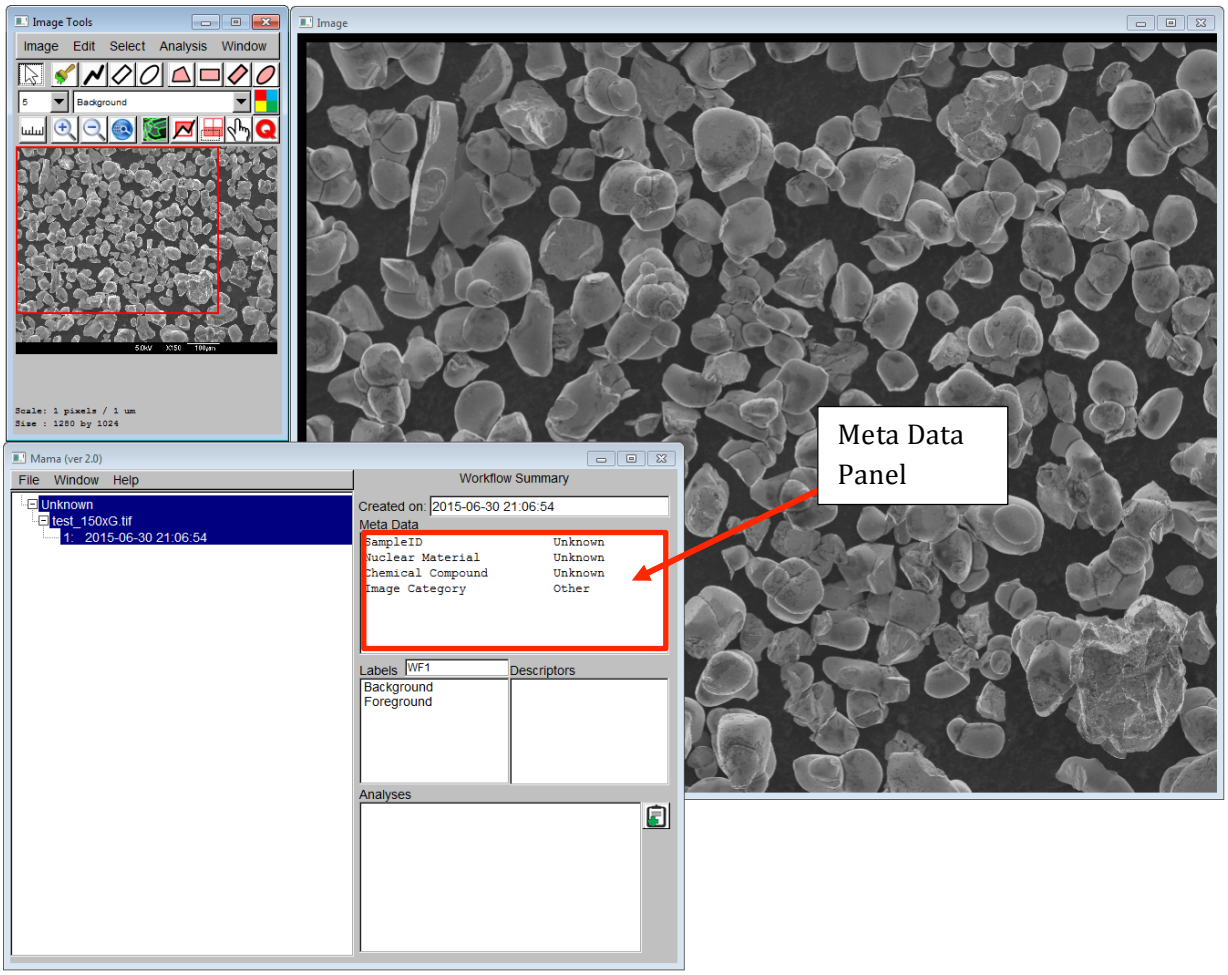

The image viewer (3) will be empty the first time you run Mama. The three windows will not appear as organized as they are in the above picture (although we are working on it!). These three windows are the main control windows, by which you will navigate the program. It is often good to take some time and arrange the windows so that they are non-overlapping. The Image Tools window (2) has most of the tools you will use to quantify the image shown in the Image Viewer

window (3). The image browser (1) lets you move between different images that have quantified previously.

## 1.2 Main Interface Windows

Typically the first thing you want to do is to open an image. You can do this from the Image Bowser **File -> New Image** or the Image Tools window **Image -> New Image.**

If you have uploaded images previously, you should see these listed in the Image Browser. The Image Browser organizes images by samples. A new image is assigned a default sample name "Unknown". You don't need to organize your images by sample if you just want to quantify and export quantification to another database. But providing the sample name is useful if you want to use multi-sample analysis tools. To tell Mama what sample is associated with the image you need to open the meta data window. You can do this through the **Window** menu item on the Image Browser or **Double Clicking** on the **Meta Data** panel in **Workflow Summary.**



The **Image Window** can be resized by standard click/drag. The magnifying glasses on the Tool Window will allow you to zoom in and out, and resize the image to the full window. If the image does not appear or is not centered, use the sliders on the side of the window. These zoom options can also be selected from the image tool bar (Image tool bar under the View menu.

The **Image Tools** window also has a navigation image that shows you the current view (in red). You can move this rectangle by clicking on the navigation image.
The **Image Tool** menu option:

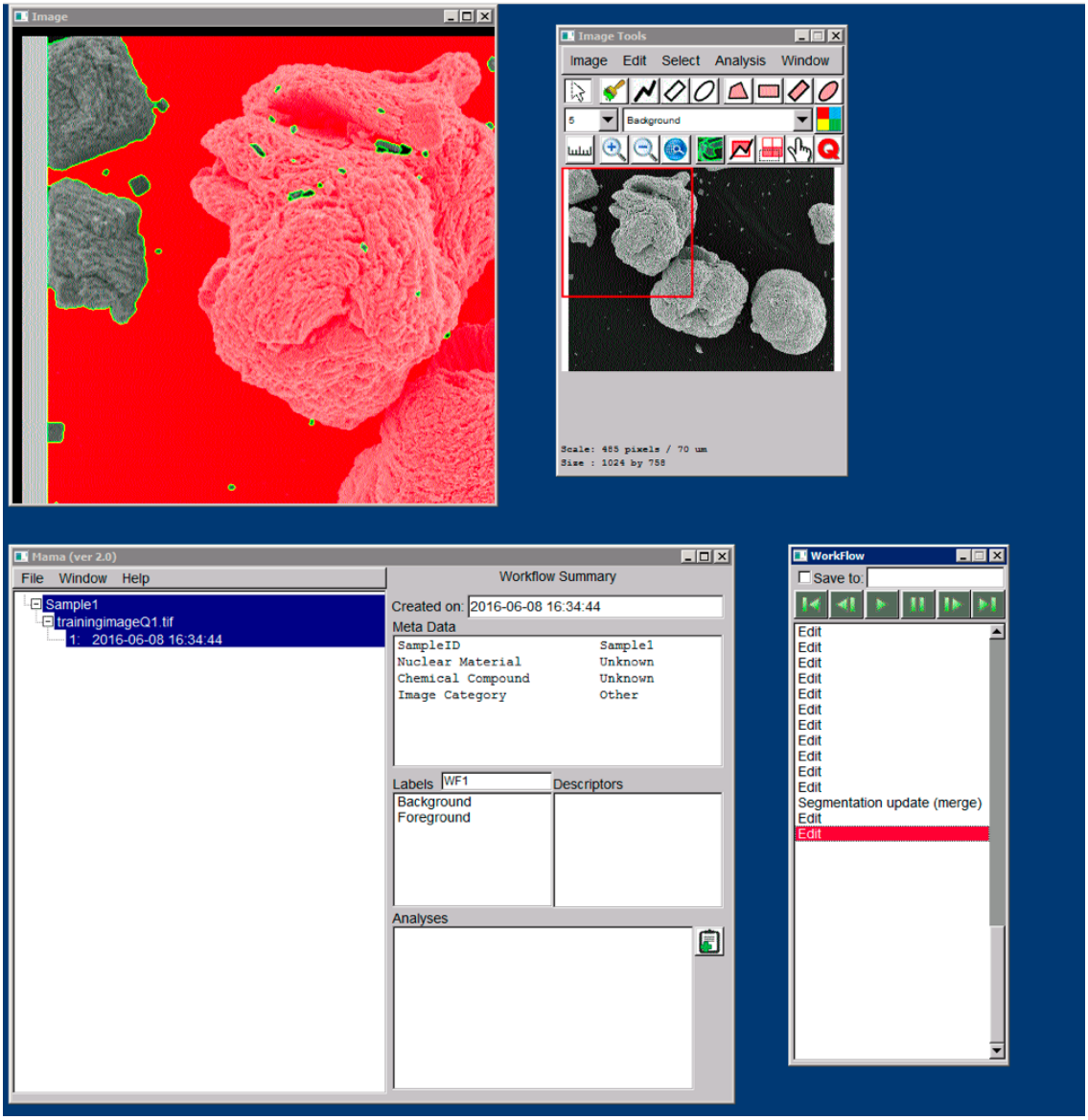
**Edit->View Options** will bring up a widget to allow you to scale the image display to a range (bottom number set) or clip the image display range (top set when 'clip' is checked.) to make annotations and segment marks easier to see. These settings do not change the actual image, but only change the display on your screen to make it easier to view. They can be thought of as equivalent to changing the brightness and contrast on your monitor to make it easier to see pictures.

Mama also has other windows that can be opened from the "Window" tab on the Image Tools or Library Browser window. Many of these windows can also be accessed more conveniently by double clicking the various panels in the **Image Browser**. We talk about some of these latter in this document

## 1.3 Workflow

The Workflows Window provides a step-by-step listing of the operations performed on an image in MAMA. When a new image is opened for the first time, an empty workflow is automatically created. Workflows are always associated with a single image in MAMA. As the user works on an image, each operation causes a new step to be created in the Workflow Window. Associated with each step is a hidden workflow image file containing the image at this step. Also associated with each step are the image metadata and any analyses performed. Workflows are saved between sessions, so opening an existing file in the Image Browser will also open the entire workflow history for that image.

To view the Workflow Window, select "Image Tool->**View->Show Workflow**" or "**Image Browser->Window->Show Workflow**". A Workflow Window will appear summarizing all actions performed on the image for the current workflow. As you perform operations on an image, each operation will be added as a new step in the Workflow Window.

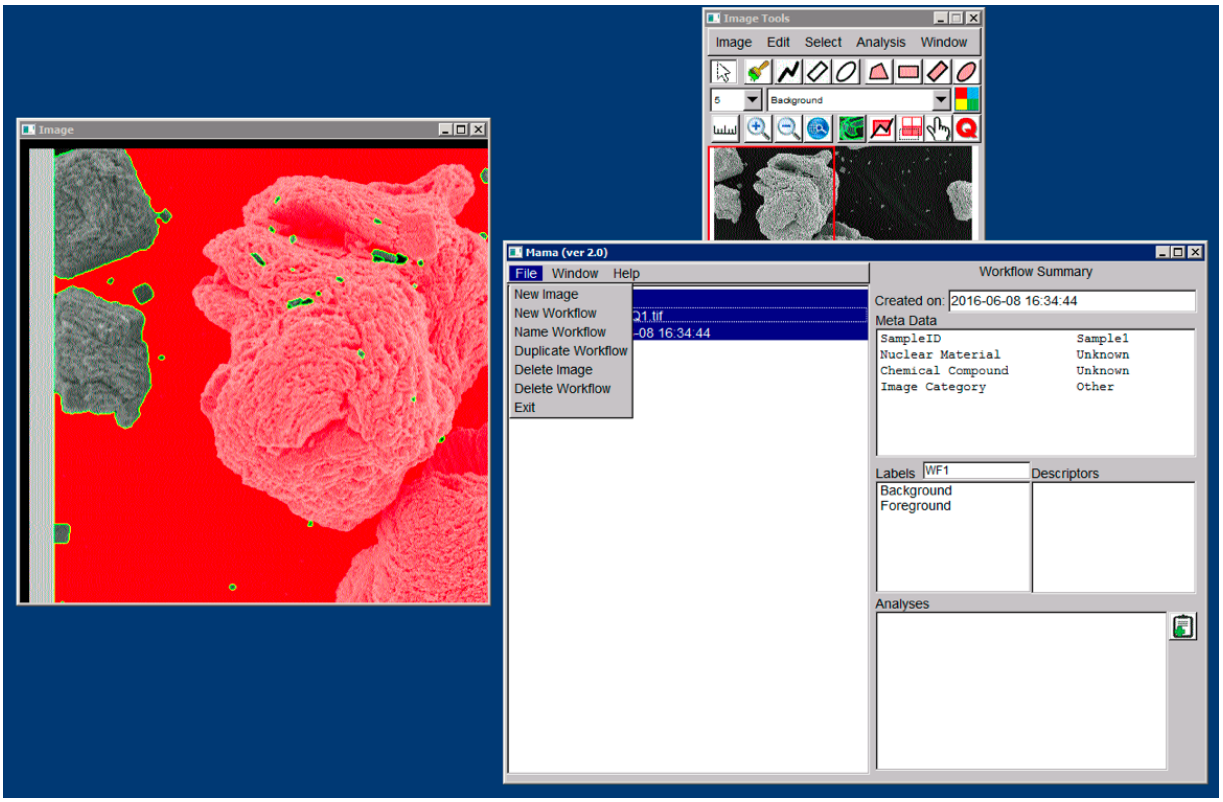



(Clockwise, from top left) Image Window, Image Tool Window, Workflow Window and Workflow Summary Window.

### 1.3.1 Workflow Management

MAMA allows multiple workflows for each image, allowing you to work with different areas of an image or to manipulate a single area in multiple ways as a way to optimize and aggregate analysis.

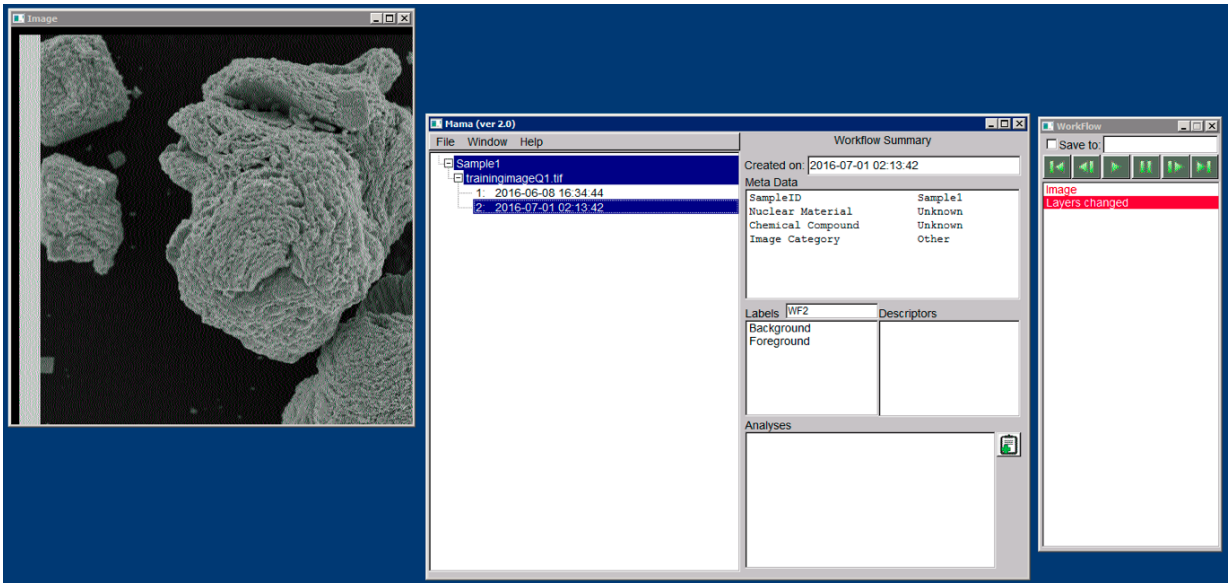
The File pull-down menu (Workflow Summary->File) gives users options to begin a new workflow, name workflows, duplicate existing workflows and delete workflows.




File Pull-down Menu

### *New Workflows*

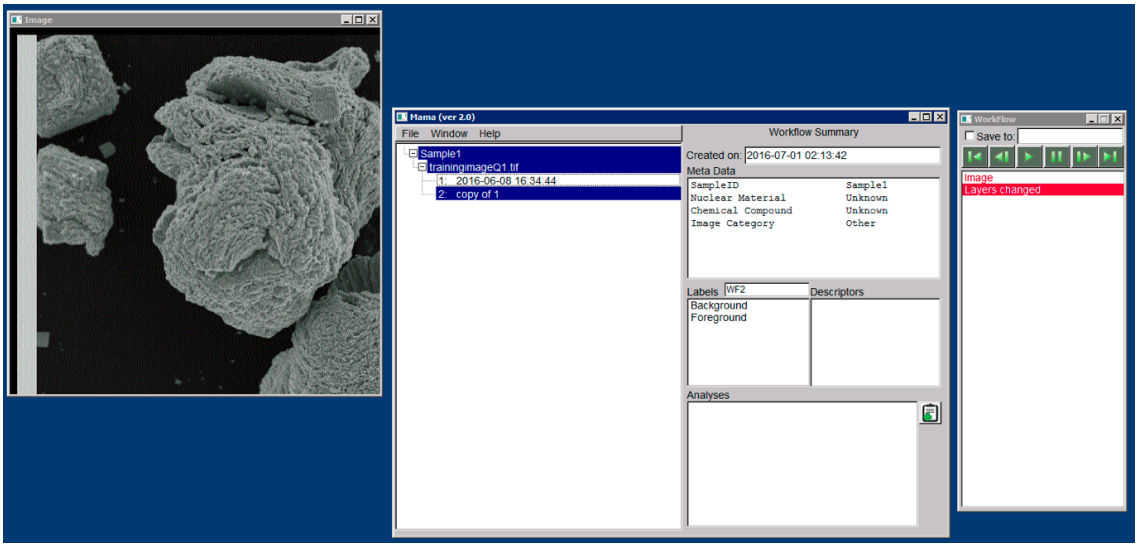
Selecting the "New Workflow" option (Workflow Summary->File->New Workflow) begins a new empty workflow from the original image. Any modifications made to the image in preexisting image workflows are not copied into the new workflow.




Creation of a new workflow

### *Rename Workflow*

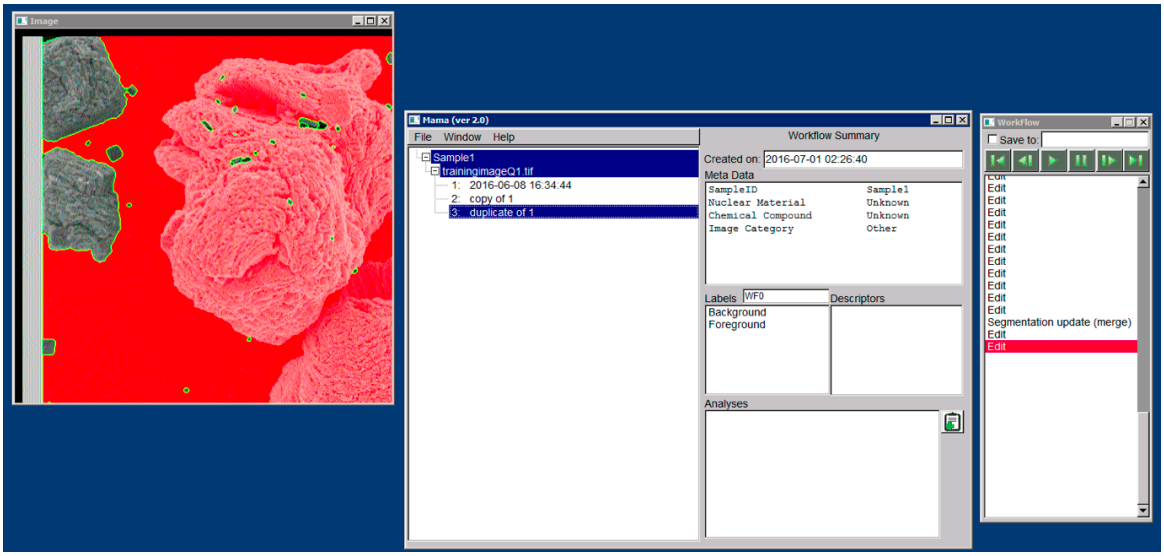
Selecting the "Rename Workflow" option (Workflow Summary>File->Rename Workflow") allows you to rename the workflow to something more understandable. By default, workflows are named with at timestamp indicating the day and time of creation. This can be confusing when managing multiple workflows and renaming the workflow to something more descriptive of the workflow is encouraged.




Renaming of a workflow

### *Duplicate Workflows*

Selecting the "Duplicate Workflow" option (Workflow Summary->File->Duplicate Workflow) duplicates the selected workflow to a new workflow. Any modifications made to the image in the pre-existing workflow are copied to the new workflow. This option is useful, for example, when you want to branch your analysis at a selected point and explore alternative methods for analyzing an image, or to backtrack to earlier step and modify your analysis.




Duplication of an existing workflow

#### *Delete Workflow*

Selecting the "Delete Workflow" option (Workflow Summary->File->Delete Workflow) deletes the selected workflow from MAMA and your hard drive. Note that the image and any other workflows remain accessible in the workflow selection tree in the Workflow Summary Window. If the workflow being deleted is the only workflow associated with the image then the image file will also be deleted. Selecting "Delete Image" option (Workflow Summary->File->Delete Image) deletes the image and all associated workflows from MAMA and your hard drive.

It is important to understand that over time workflow/image files can take up a considerable amount of hard disk space on your computer. Since these files are quite large (each step in a workflow contains a image file and pointers to metadata and all analyses performed in the workflow to that step), you may find that your available disk space becomes limited over time. **Only delete workflows/images from within MAMA; selectively deleting them directly from your hard drive can corrupt MAMA's internal database and cause the program to behave in unexpected ways.** For more information on the location of image data/workflows, see Appendix B, Data Storage.

### 1.3.2 Workflow Traversal

#### *Viewing a Previous State*

Selecting any step in the Workflow Window will cause the Image Window to display the state of the image at the time that operation was performed.

**While it is possible to apply an operation to an image when displaying a previous state, this is strongly discouraged**. MAMA does not create a branch but will attempt to apply your change to the step selected and to all subsequent steps in your workflow. This can result in highly unpredictable behavior including early termination of program execution and the corruption of the workflow. **Traversal of the workflow should only be done for viewing the effects of previous modifications to an image.**

**Best Practice: Apply modifications only to the last step in the workflow.**

If you need to undo steps in the workflow, use the undo command (Image Tools ->Edit->Undo Edit), which will undo single operations beginning with the last operation performed. There is no limit to the number of undo operations that can be performed. To redo an edit that has been discarded, select (Image Tools->Edit->Redo Edit).

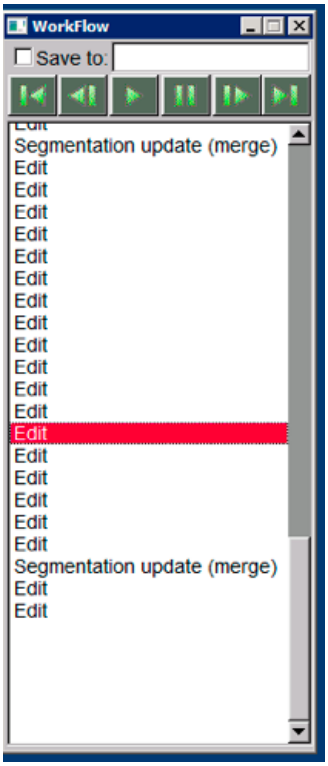
If you find you want to apply a different analysis to a previous step in a image but want to maintain your current workflow, duplicate that workflow (Section 1.3.1) then apply Undo to the duplicate workflow until you reach the step you need to modify.

If you want to branch a workflow, use "Duplicate Workflow" as described in Section 1.3.1

#### *Workflow Playback*

MAMA allows you playback your workflow beginning at a] user-selected starting step and continuing through the remaining workflow, much like playing a mpg movie. The image displayed

in the Image Window changes at each step of the workflow allowing you to view the changes that have occurred to the image while you have worked with it. There are controls at the top of the window for single step forward, single step backward, play entire workflow, advance to beginning, advance to end, and stop. Note that the SAVE TO selection at the top of the playback tool is non-functional.

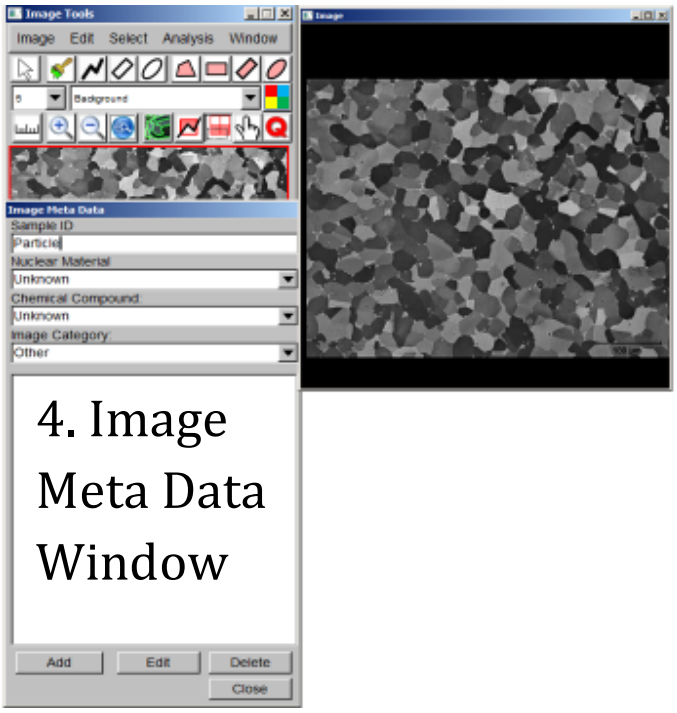



Workflow Window with playback buttons at the top of the window

## 1.4 Image Meta Data

The MAMA Meta Data provides a lightweight database for meta-data associated with the image. This is not supposed to replace your existing sample database[1], but does let you provide meta-data for the multi-sample comparison tools that will be discussed in Section 3. Only four fields are specifically set: the sample ID or name, the nuclear material, the chemical compound, and an "image category." You can add edit and delete custom Keyword / Value pairs on the lower panel (using the add/edit/delete buttons). This is useful for storing image specific information associated for example, with an experiment set (e.g. click add, then enter attribute "Temperature (Celsius)", click ok, then enter value "350" and click ok.

[1] If you would like us to try and automatically populate metadata from your existing database please contact us at mama@lanl.gov

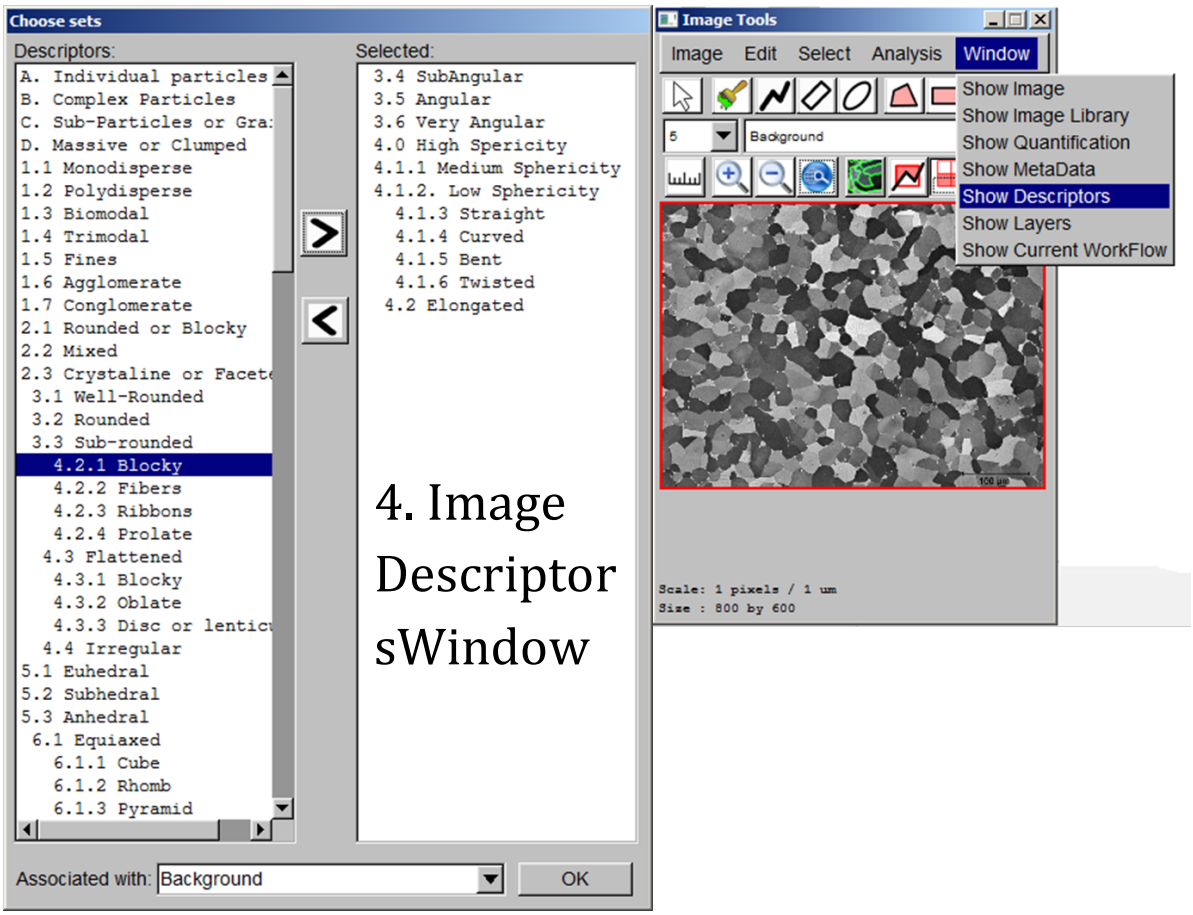



The image category has seven descriptors for categorizing images into groups that are likely to have similar analysis workflows. This will simplify data exploration/ visualization and other comparative data analysis that we plan to implement. Below this is an image category is the image size (in pixels), number of channels (1 for a gray scale image) and resolution in pixels per micron. This value will default to 1.00 until the ruler is set (Section 1.6.1).

**Development Notes:** 1) in the current release, scale numbers cannot be used in the data visualization tools. We plan to add this capability in the near future, since it's often useful to compare images at multiple scales. 2) A prototype for complete chemistry metadata databases entry and sample history entry was developed, but was not included with this release. This was a design choice made for a number of reasons: few users in beta testing wanted to entry meta data beyond the minimal fields necessary, extensive chemistry data was not needed at this stage of morphology software development, and most users had existing complete chemistry databases that could be more effectively used by providing database integration tools rather than a separate duplicate database. We are actively working on improving the sample processing history meta data window and fields to align with NTNFCs formatting, and to contain information that is needed for sample analysis and comparison that is missing from typical chemistry databases.

## 1.5 Image Descriptors

Another useful windows related to meta-data is the Image Descriptors window:




This window provides a set of standardized terms for describing the image content. Users can simply click the appropriate characteristics for their image. Note, that the "Associated with" at the bottom of the window. This is to enable you to select terms for specific sub-parts of the image. We will explain this more once we have discussed how to label parts of the image (Section 1.8).

**Development Notes:** the standard lexicon is open to discussion and modification by the community. If you have input/opinions on this part of the program please email mama@lanl.gov. Additionally, the terms listed here are the original listing and organization of terms that we developed. The list has been modified somewhat since this original term set, and has been sent out (along with term definitions) to all people who expressed interest in the MAMA software. We will update the term list and structure in the software to the next consensus listing in a future release, after we have given people time to provide feedback.

## 1.6 Image processing-getting started

MAMA has many tools to aid in processing your image for quantitative analysis. On the Image tools window, these tools are grouped under the **Analysis** menu. Exactly how you use the tools will

depend on your image and analysis needs. We plan to host tutorials on many of these features as the project progresses. This tool group includes the following:

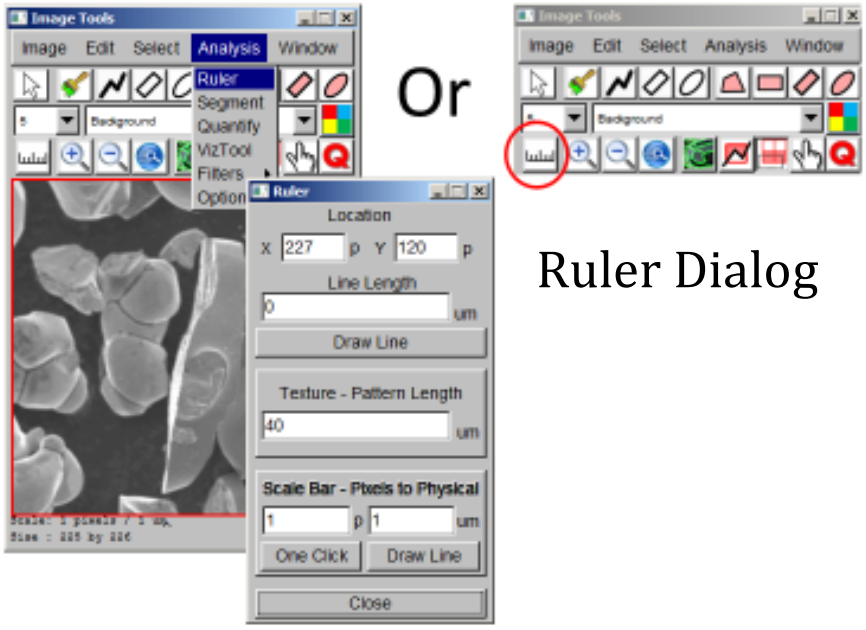
**Ruler:** for establishing pixel to a physical measurement (section 1.6.1)

**Segmentation:** for separation individual particles or grains before quantification. (Section 1.7)

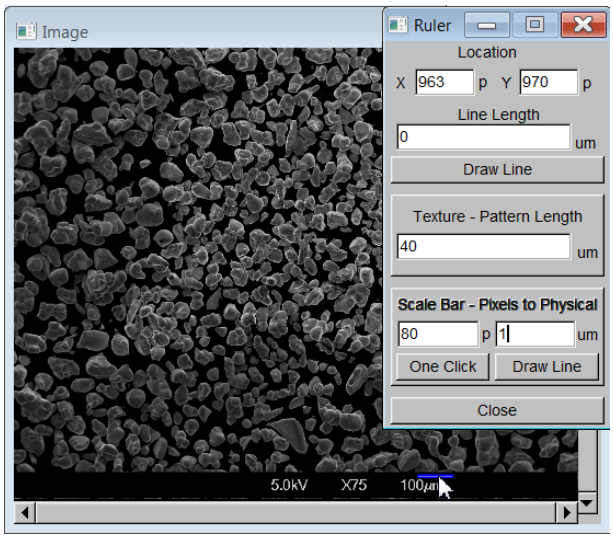
**Quantify:** these are tools used for particle quantification (Section 1.8) and texture quantification (Section 1.9). These tools require that a segmentation has been previously performed.

**Filters:** for manipulating and correcting image-wide illumination artifacts or creating a temporary image for improving segmentations (Section 1.6.2)

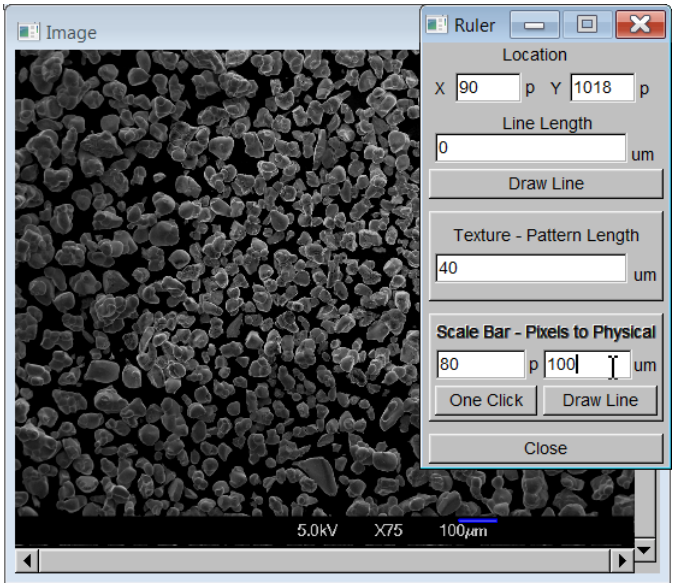
### 1.6.1 Image processing-Ruler




On the image tools window, under the processing tab is the rule selection. This also a tool bar icon for the ruler. The Ruler is what transforms quantification results from the pixel domain into physical units. There are three ruler function grouped into this window. The most important part of the dialog is at the bottom: **Scale Bar – Pixels to Physical**. This dialog is used to enter the actual physical scale from the image. This is typically done from the scale bar that is overlaid on most SEM images. There are two buttons that aim to make this easier for you. **One Click** is for cases where the scale bar is a single uninterrupted line (as it is in the example image). In this case, you can press **One Click** and then click on the scale bar line in the **Image Window**:




If successful, MAMA will highlight the scale bar line in blue and then automatically populate the left field (**P** for Pixel) on the Ruler. You can now manually type in the number of physical units into the **µm** field:




You can now **Close** the Ruler. Quantification results for this image will now be reported in physical units. Note you will typically enter different data into the ruler for each image in the database.

If the One Click option does not work then you can use the **Draw Line** option that lets you draw a line on the image using two mouse clicks. Click (and release) at the start of the scale bar line, then move the mouse to the end of the scale-bar line and click (and release) again. (Note: for increased accuracy, you may need to zoom into the scale bar. This can be with the  buttons- zoom in, zoom out, and full window scale.) If successful this will have a similar effect to the **One**

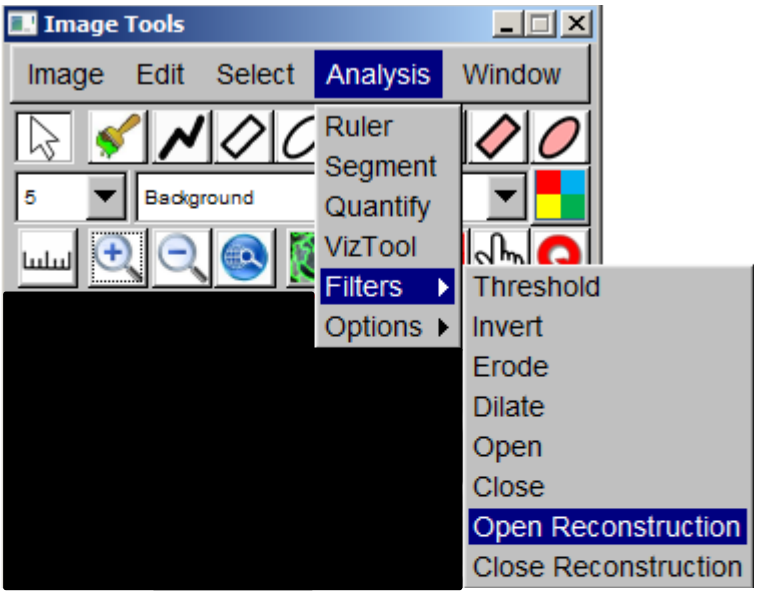
**Click** tool in that it will automatically populate the **P** field based on how many pixels long the line is, and you then enter the length in microns.

This same functionality is also part of the draw line button in the top of the **Ruler** Dialog. This is used for measuring objects in the image by hand. However when you press **Draw Line** it will not auto-populate the scale bar – it is meant to help users measure content in the image once the ruler has been set (and will not show a result unless the ruler is set.)

The middle part of the dialog is for setting the scale parameter for quantifying texture. This will be discussed in Texture Analysis (Section 1.9).

### 1.6.2 Image Processing-Filters

MAMA provides a number of standard image processing routines that can be used to reduce noise, or enhance features of interest prior to quantification. These are accessed through the **Filters** menu option. NOTE: Use of filters is not necessary for a lot of images, and is typically used by advanced users.




Some of these filters are completely automatic, e.g. **Illumination Correction**, while others allow users to make specific choices, and adjust thresholds. They can be used to correct image artifact or to produce a temporary image layer that can be more easily segmented. The filtered image can be used for segmenting, and then (using the image layers window, Section 1.11) the original image can be selected for analysis & the filter image turned off.

**USE NOTE:** Most of these filters will have an effect on some of the texture quantification calculations, if the resulting 'filtered' image is used for quantification. Use filters with caution, and perform quantifications (in most cases) on the unprocessed image. There are no catches built into the software at this point to prevent a user from altering an image using these filters, and then using the altered image for quantification.

The following filters are implemented:

**Illumination correction:** this will even out the average illumination over an image. This is useful when image acquisition effects produced shadowing over part of the image.

**Threshold:** This will pop up a small window with a histogram of the pixel intensity and a slider bar. Thresholding will convert the grayscale image to a binary image. The slider can be used to determine the pixel intensity level at which to split the image. At one extreme, pixels at almost all intensities are converted to white; at the other extreme, pixels at almost all intensities are converted to black. The optimal threshold is likely somewhere in between—often this can be determined by looking for the troughs in the histogram, so that groupings of pixels of similar intensity are split from other groups.

**Invert:** This alters the original color on the pixels in an image to their inverse.

**Erode & Dilate:** These filters are mathematical operators that apply a structuring element to an input image to generate an output image. Dilation causes bright regions in an image to grown, while erosion causes the bright areas to diminish. Both these filters have sliders to preview the effect of the filter at different levels. These filters can be useful for obtaining segmentations in which there is a large texture gradient. They are very useful for removing noise and isolating individual elements. Note that these filters will undoubtedly change the image texture and may affect particle size. If they are used for obtaining segmentation, the original image must still be used for quantification.

**Reconstruction:** Reconstruction will open up a large window with 2 sets of radio-buttons and a slider. Reconstruction effectively acts as a 'smoothing' function on grayscale images. This can make segmentation far easier, and often with better results than thresholding. The reconstruction can be set to use round, linear, or area features, and use light, dark, or light/dark in relation to neighboring pixels. A slider then sets the amount of reconstruction. We suggest experimenting with these options to understand what they can do and how they can help give an easy-to-segment image.

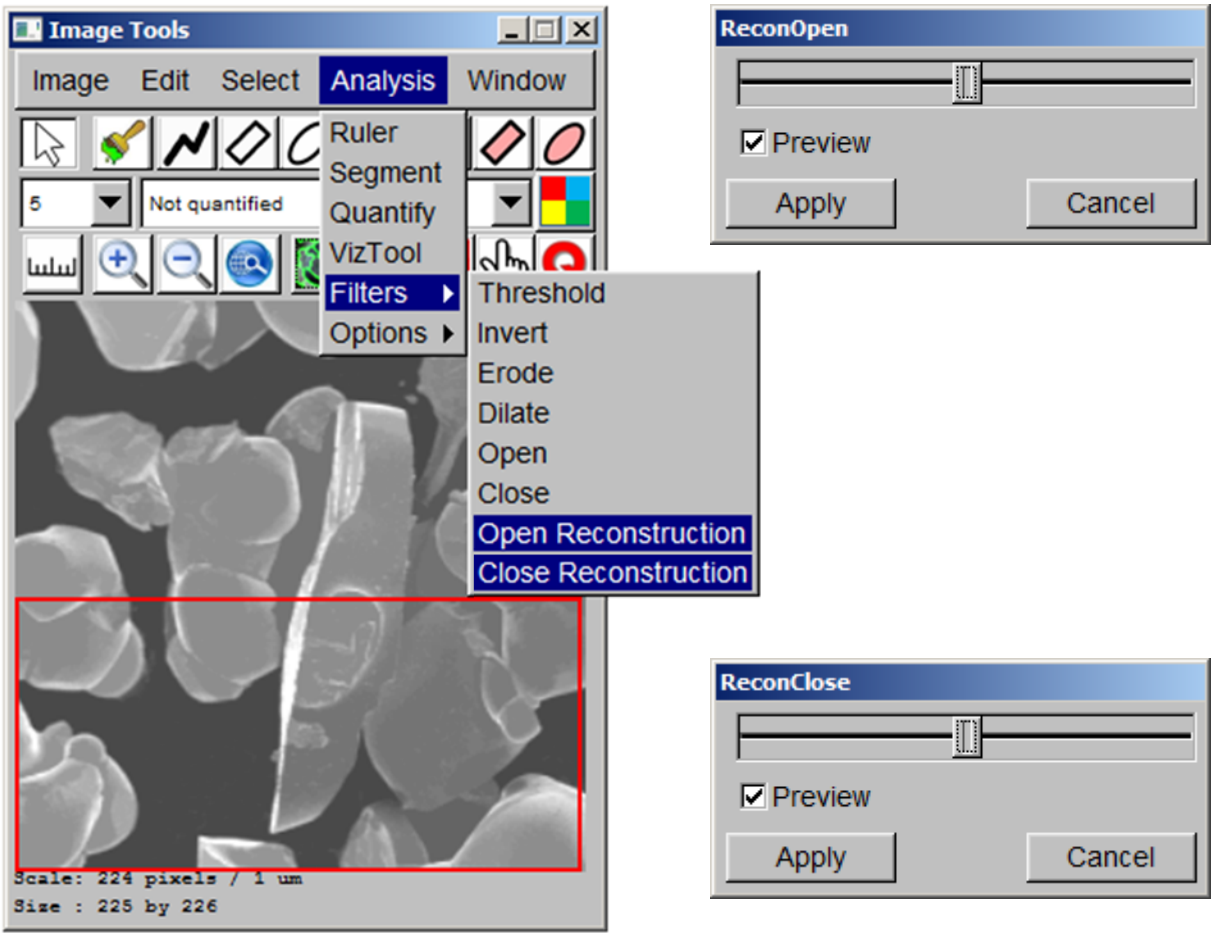



Image Artifacts: will automatically segment out the entire image metadata information bar region that appears at the bottom of many images. This region can then be labeled as background (or other label), as described in the following sections.

## 1.7 Image Segmentation

Mama provides a large number of tools to help provide accurate image quantification. Image quantification falls into two main categories: **Particle Analysis** and **Texture Analysis**. Both categories use the same basic workflow, which have three main steps: ***1) segmentation, 2) labeling and 3) quantification***. These tasks can be used iteratively to obtain and refine the segmentation and quantification.

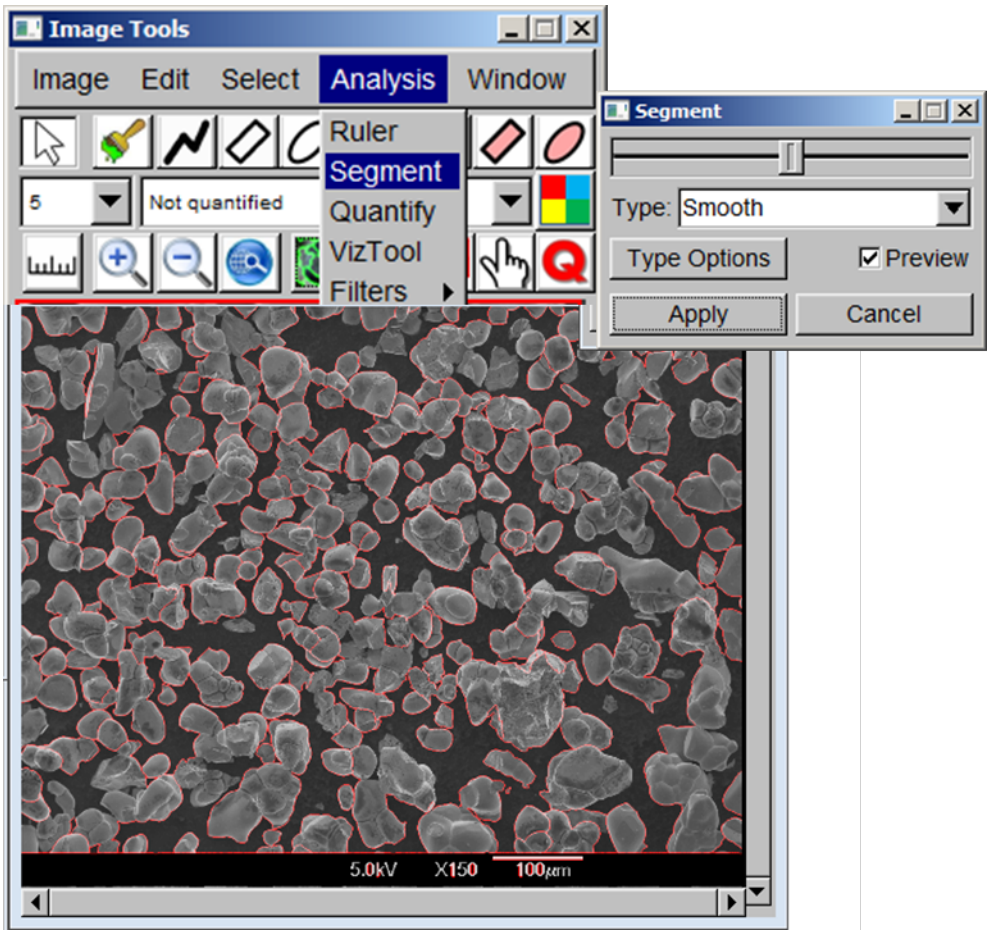
Much of the subjectivity (and variability) during quantification occurs in steps 1) and 2). MAMA provides a number of tools to help users in these tasks. In combination these tools can handle a variety of image types and backgrounds. It is worth spending some time getting familiar with how these tools work together conceptually.

**USE NOTE:** If you only want to quantify the texture of a region in an image (i.e. the parts of the image that are show a agglomerated material, the internal parts of a particle image at high

resolution.) go to Manual segmentation with Annotation section (Section 1.6.6). There is no need to try to do automated segmentation of individual particles.

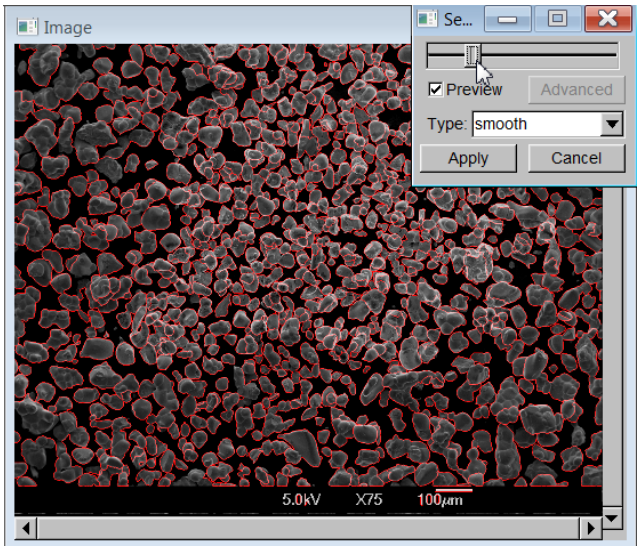
### 1.7.1 Semi-Automated Segmentation

Segmentation is the process that splits an image into regions (also called segments). Typically a region would correspond to a real-world object (e.g. a particle, or grain), but this is not always the case (e.g. in the case of texture). The automated segmentation routines predict an initial segmentation based largely on *edges* within the image.




4. Segment Dialog

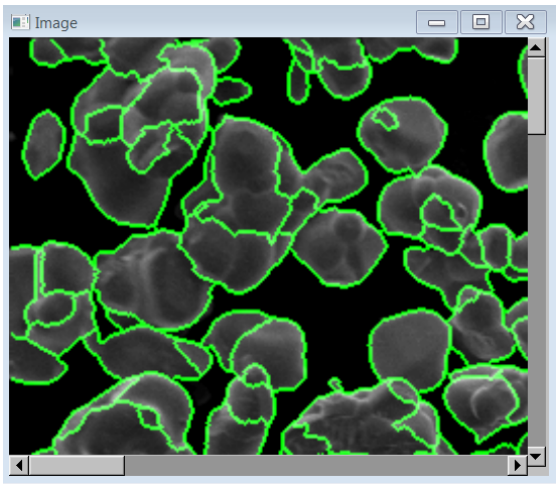
A **Segment** Widget is displayed which allows the user to move a threshold slider. This slider selects the scale and typically changes the number of separate particles predicted by the algorithms. Try and select the threshold that is closest to the desired result (as indicated by red boundaries drawn around predicted particles). Press **Apply** to accept.

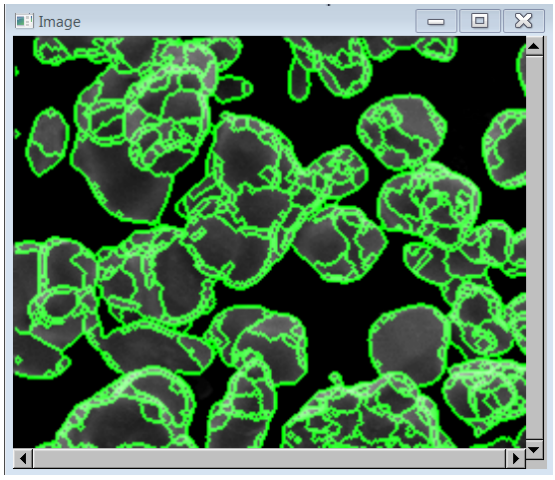



How accurate this segmentation is will depend on your image. In some cases, with well-separated particles, the initial segmentation may be enough. Images with confusing or weak edges (segments that have widely varying pixel intensities in each segment, or images with segments that have little difference from neighboring segments.) may require some pre-processing (with filters inside the program or with any photo-shop type program outside this software) in order to obtain the most accurate segmentation.

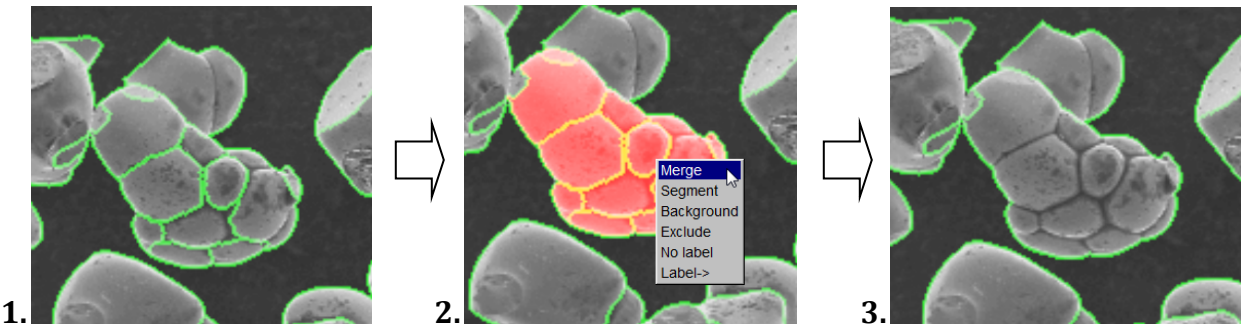
### 1.7.2 Segmentation Type

One of the main factors that affect the segmentation performance is the scale of the objects of interest with respect to the resolution of the image. The segmentation **Type** gives you some control over this aspect of the problem. The default option is **smooth** and it works well for medium sized particles and grains. If you are interested in segmenting smaller objects than the slider allows (e.g. to quantify inter-grain inclusions), or larger objects than are initially segmented (e.g. to quantify macro-structures), then try changing the **Type** parameter. Note, for more detailed **Types**, the computation can take some time, so please be patient. Eventually you should be able to move the slider as before, but this time, you should notice the segmentation focuses on smaller objects. On the left below, we see the default (smooth) **Type**, and on the right we see the **Detail Type**. In this image, the smooth type is clearly a better choice.



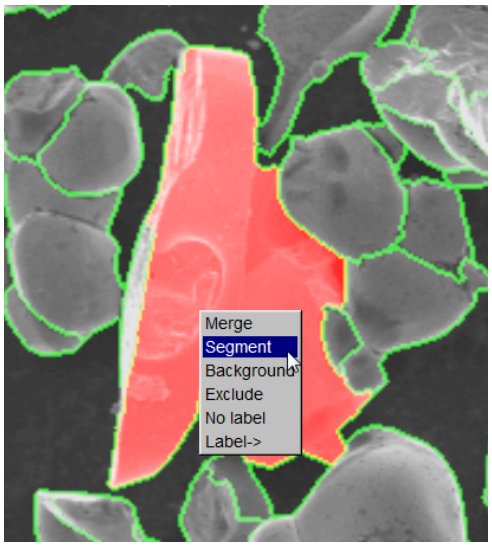




### 1.7.3 Merging Segments:




If the result is over-segmented (Left image), users can **left-click** segments to select (highlighted in red). After all segments have been highlighted, **right-click** to see options (Middle Image), select **Merge** and the segments will be combined into one segment (Right image).

### 1.7.4 Threshold Refinement & Multi-step Segmentation

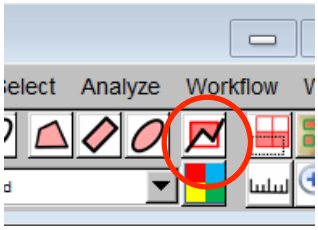


If the result are under-segmented, a user can **left-click** the segment to select (highlighted in red). Then **right-click** to see options (Left image above), select **Segment** and Mama will let the user adjust the segmentation threshold locally to the selected region. Press **Apply** to accept the new segmentation.

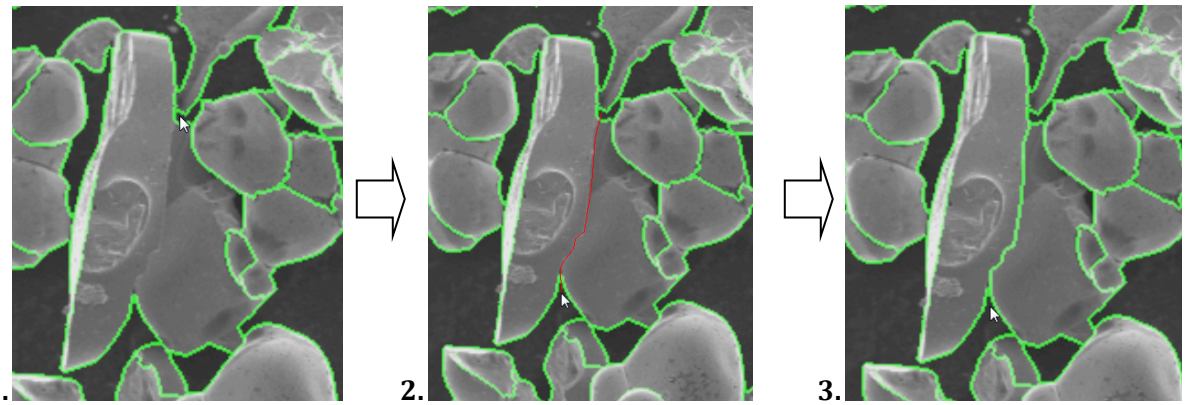
Note that this multi-step segmentation can very effectively be used to segment some difficult images—an initial segmentation can be obtained that at least segments some regions that need to be separated (for example, a large part of the background from the particles, or one type of particles accurately segmented from the background.) These resulting initial set of segments can then be individually further segmented as needed. This can sometime save having to do extensive merging on many segments. Alternately, for an image in which there are different regions requiring different levels of segmentation, the manual segmentation tool with annotation tool (Section 1.7.4) can be used to initially sub-divide the image into regions for segmentation, and then each resulting segment can be further segmented. (If there is an image that presents segmentation challenge, please send it to us; we can try to show how to best use the tools to obtain segmentation, and then send the 'workflow' back to you.)

### 1.7.5 Manual Splitting

Sometimes adjusting the threshold does not improve the segmentation. In this case, it is often faster for a user to manually delineate the particle. Mama provides a line drawing tool for this purpose that can be accessed on the toolbar, as shown in the image on the right.




After pressing this button, the user should **left-click** once in a region close to the boundary but outside of the segment of interest. This is shown on the left below.

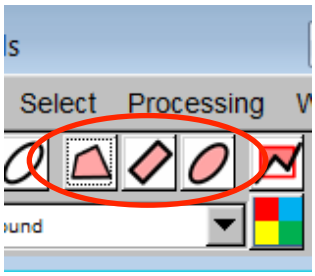



The second step is to define a boundary, with multiple **left-clicks**. This is shown in the middle image above (red line). The final step is to **double-click** outside of the segment of interest as shown on the right image above.

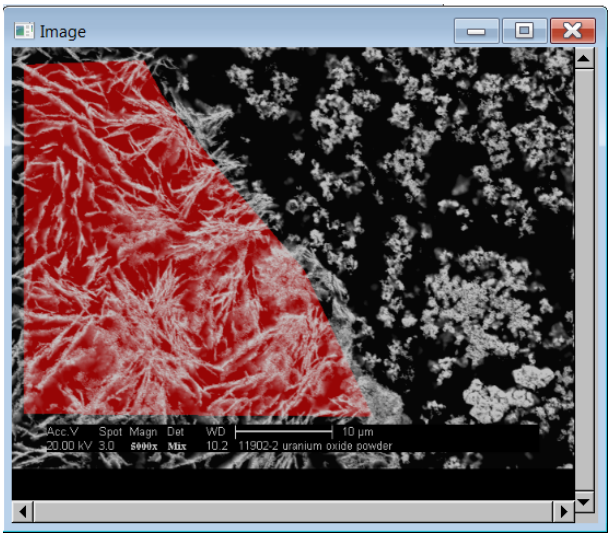
**USE NOTE:** instead of selecting this tool from the toolbar, it can be engaged by holding down the left CTRL. You should hold it down throughout the split until after you have double-clicked.

In order to achieve the highest consistency between users, automated segmentation should be used whenever possible, with manual splitting as a last resort.

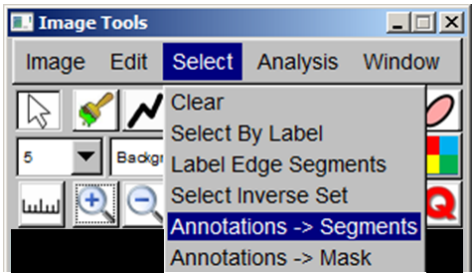
### 1.7.6 Manual Segmentation with Annotations

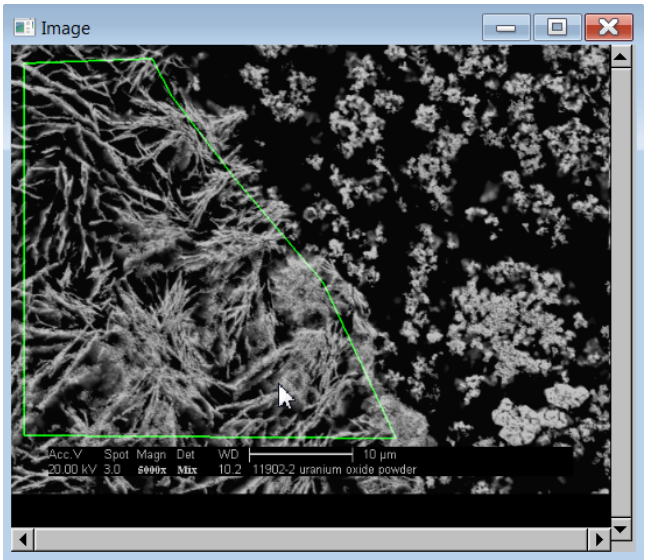



Sometime the automated segmentation is just not able to get the content you are interested in. Other times, it's less about image content than it is about defining regions of interest, or regions of good image quality for further segmentation or analysis. This is typically the case for images in which there are not clearly defined particles, and user are interested in the overall texture of the bulk mass, or when quantification of the texture of a single large particle at high resolution is desired. In these cases, you can use the annotation tools to draw user-defined shapes on the image, which you can translate into regions or segments. Typically you will use the **filled** polygon, rectangle and ellipse tools:




In the image above the filled polygon tool (double click to finish the polygon) was used to identify a region of fibrous agglomerations. Multiple regions can be defined using multiple annotations. To transform the annotations into regions (or segments) select from the menu or use the ToolBar button.






The green lines above indicate that the image now has two segments – the region of interest, and the region around it.

Note in this image, a second type of particle agglomeration exists—a user may want to capture these agglomeration regions using the polygon tools rather than direct segmentation, and then label the segments as two types of agglomerations, and label the rest as background. (See next section for how to label.) There is no limit on how many different regions can be identified.

**USE NOTE:** If you already have segmentation, selecting **Annotations -> Segments**, will combine the annotations with the current segmentation. **Segmentations cannot overlap**-all segments inside the annotation region will be combined into the new annotation region segment. Segments on that are split by the annotation boarders will be split. If you want to segment in different ways for different quantification needs, which would result in segments that overlap, see the next section.

### 1.7.7 Different segmentation on the same image:

If you wish to segment different ways to quantify different features, there are 2 ways to do this.

1. If you want segmentation for quantifying particle size/shape features, and a segmentation/annotation for texture, simply do them as separate workflows on the same image. If you wish to use these quantifications in multi-sample analysis (Multi-Sample features, Section 4), the software will find the most recent workflow with the correct type of quantification. Note that this means, do not perform a particle quantification in the workflow created for texture, and do not perform a texture quantification in the workflow created for particle quantification.
2. If there are multi segmentation needs that are not separated into a single texture set and a single particle set, import the image a second time to perform different segmentations. You can export an initial segmentation from one image and import it into another if the starting segmentation is the same. Note that a single segmentation, but with the segments having different labels, can be accommodated with labeling (See next section.)

**Development Note:** we do plan on updating this feature, as part of a larger update for multi-annotation (multi-particle, multi-label, multi-workflow etc.) analysis using the multi-sample tools. This will allow users to name and select workflows for analysis more easily, so that an image only has to be added once, and then any number of workflow can be created and selected for comparison using the Multi-Sample tools.

## 1.8 Labeling

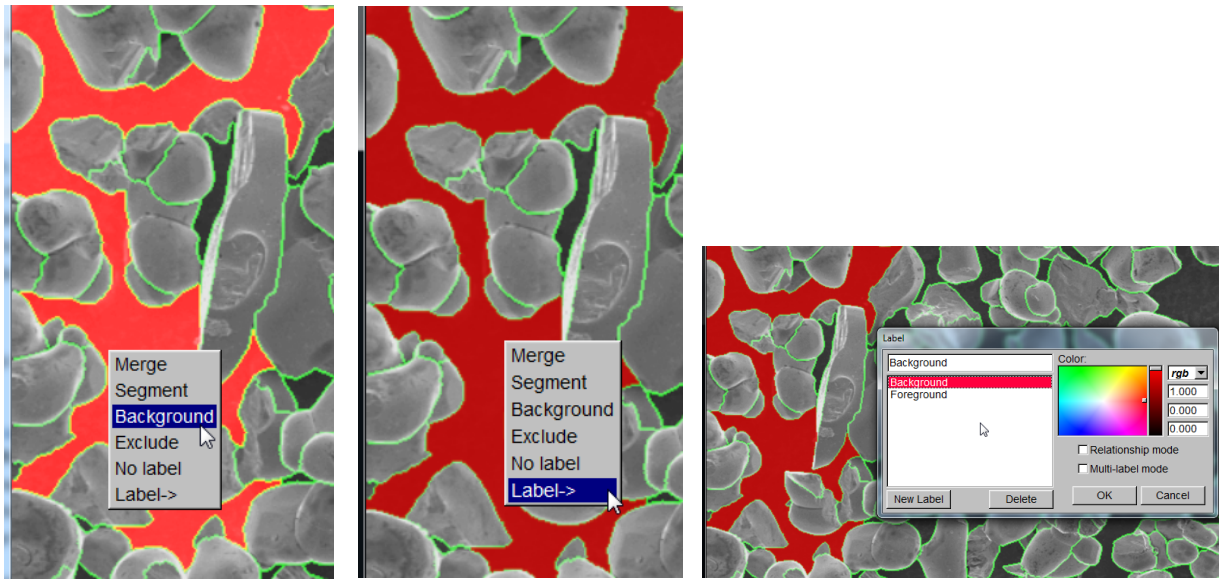
Once the user is satisfied with the initial segmentation, they will typically be interested in defining which parts of the image are of interest**. In almost all images, this will involve labeling the background and instrument- generated scale bars that are written onto the image (as in our example), that we would like to exclude form quantification**. However, for complex materials, a user may wish to use multiple different labels for different image/material features. Mama provides a number of tools to assist in this process.

**USE NOTE:** The software cannot distinguish particles from background or artifacts without the user. The user must label the background in order to exclude it from quantification. Artifacts can be eliminated by hand (selecting the region by hand, and labeling as background) or by selecting them using the Processing->Filters-> Instrument Artifacts filter and labeling as background.

**Development Note: we** do plan on updating this feature, as part of a larger update for multi-annotation (multi-particle, multi-label, multi-workflow etc.) analysis using the multi-sample tools. This will allow users to use the annotations labels they created independently in the multi-sample analysis using the Multi-Sample tools.

### 1.8.1. Segment Selection and Labeling

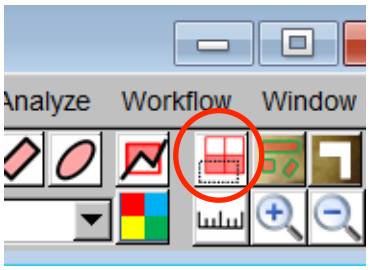
The user selects segments by simply **left-clicking** (segments become highlighted in red). Selected particles can be assigned a label with a **right-click.** Selecting **Right-Click -> Background** assigns a default label to the segments, as shown on the left in the figure below. However, **Background**, is just one example of any number of user-defined labels that could be associated with the segments. Selecting **Right-Click -> Label** (Middle image below) opens the **Label Widget** (right image below) that allows users to customize (add or rename) the label name and color associated with selected segments.



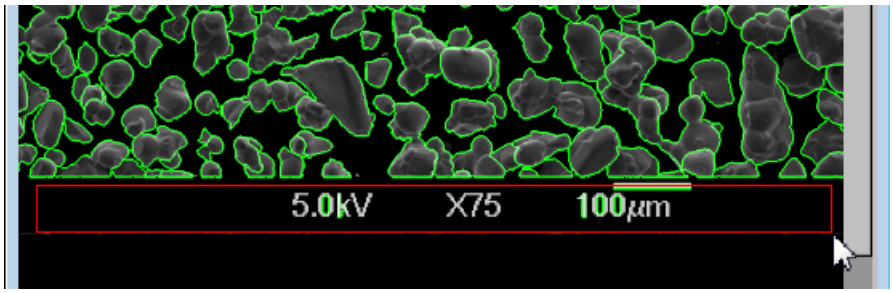

Mama also provides a number of specialized tools targeting segment-labeling problems that often arise during quantification.

### 1.8.2 Selecting groups of segments with a rectangle

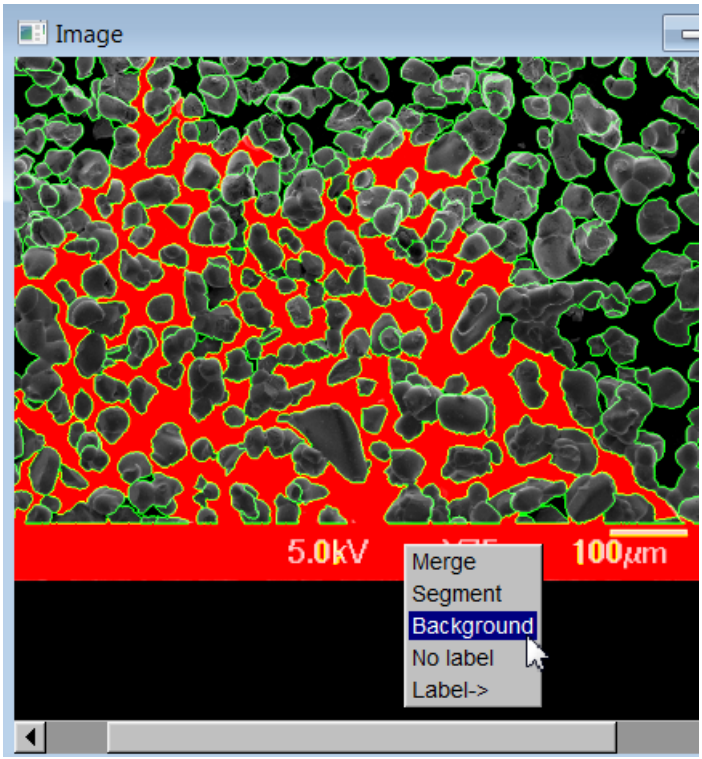
Mama provides a tool to select segments that overlap with rectangles. This tool is selected from the toolbar as shown in the image to the right.




After selecting the user can drag a rectangle over the region of interest. **Press and hold** the **left-button** to drag the rectangle. Release to finish. A typical result (before Release) is shown in the image below.



Mama selects all regions that touch the rectangle. The user can then **right-click** to assign labels.




In the image above, the user selects **Background** to indicate that the scale bar, and connected regions are not features of interest.

### 1.8.3 Label Statistics

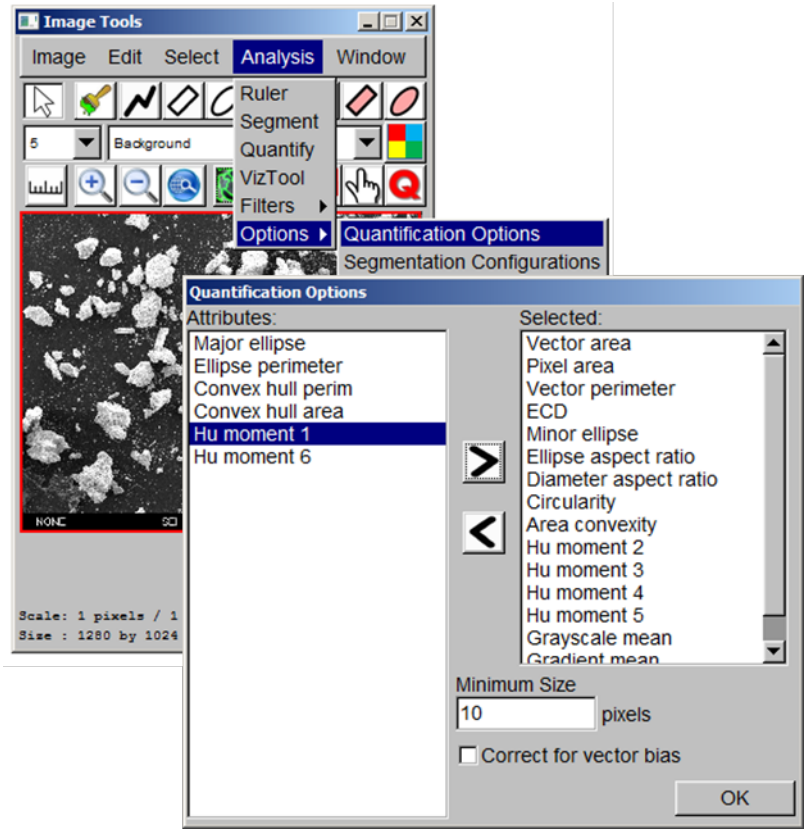
The amount of space in an image the represents the different labels can be displayed from the Image Tools menu:

**Analysis→Label Statistics.**

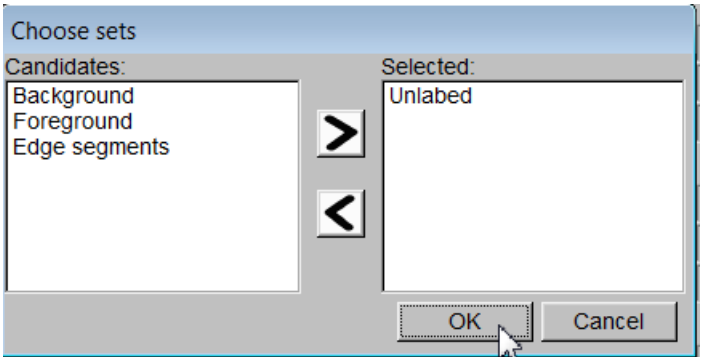
This will display window with the labels used, the area covered by each label (in $\mu m^2$), and the fraction that represents. These can be exported for comparative analysis using the export button on the label statistics window.

## 1.9 Segment Quantification

Given a segmentation (and labeling) of particles, the next typical step is to calculate attributes for each particle. You first open the Particle Quantification window.




Press **Calculate** and you will a **Choose sets** widget will appear, which allows you to specify which labels (or sets) of particles to consider.




We have found that we typically keep the content we are interested in **Unlabeled**, and this is the default choice. Move labels back and forth between the labels in the image (on the left) and the labels that will be analyzed (on the right). Press **OK** to continue with analysis.

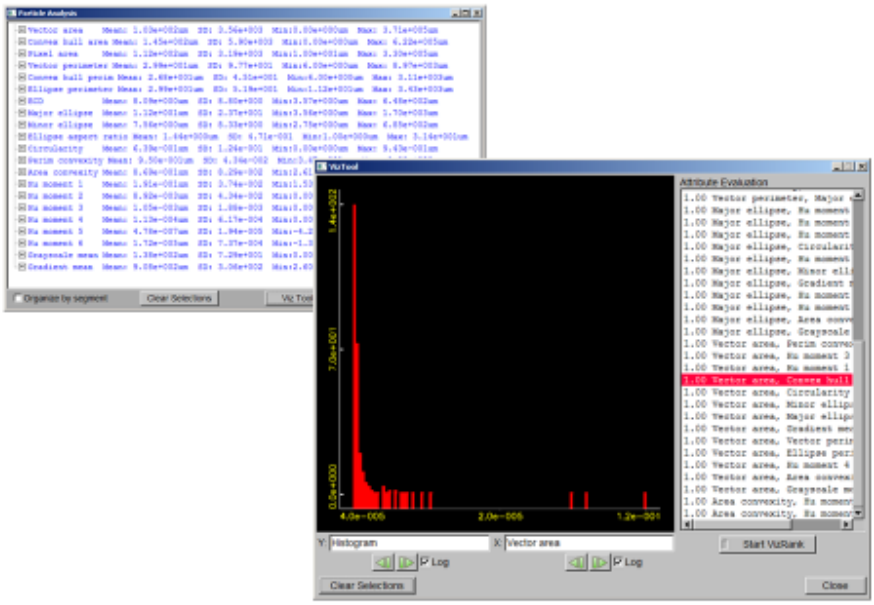
Depending on the number of particles, this step can take some time. Eventually, the table will be populated with statistics as shown in the figure above. Details of what attributes are calculated and how are provided in the appendix at the end of this document. The table shows aggregate statistics, such as the mean and standard deviation, calculated over all unlabeled particles.

The **Export** menu options allow users to specify a text file, and the particle analysis will be saved in comma-separated form suitable for import into graphing programs such as Excel. This will be the listing of the attributes for all the particles selected for quantification, not the summary statistics.

**Development Note: we** do plan on updating this feature, as part of a larger update for multi-annotation (multi-particle, multi-label, multi-workflow etc.) analysis using the multi-sample tools. This will allow users to export the annotations labels they created with the particle attribute.

### 1.9.1 Interactive Labeling and Grouping

After particle analysis, you can preview the particle distributions for each attributes by pressing the **Viz tool** button in the Analysis drop down menu. This opens the **Histogram** widget shown on the left below. The default x-axis scale is linear, however sometimes a log scale is preferable. The user can toggle between the linear and log scales by pressing the buttons on the lower left of the widget, as shown in the middle image below.



With the histogram widget you can select groups of particles in the image window that have attributes within certain ranges. Hold down the **left-button** on the plot window and **drag** the white selection region over the range of interest, as shown on the right image above. Particles whose attributes are within the selected range will be selected on the main image window. This is useful for quickly labeling sets of particles. For example, a user might:

1. Press **Plot** for the “Particle Area” attribute.

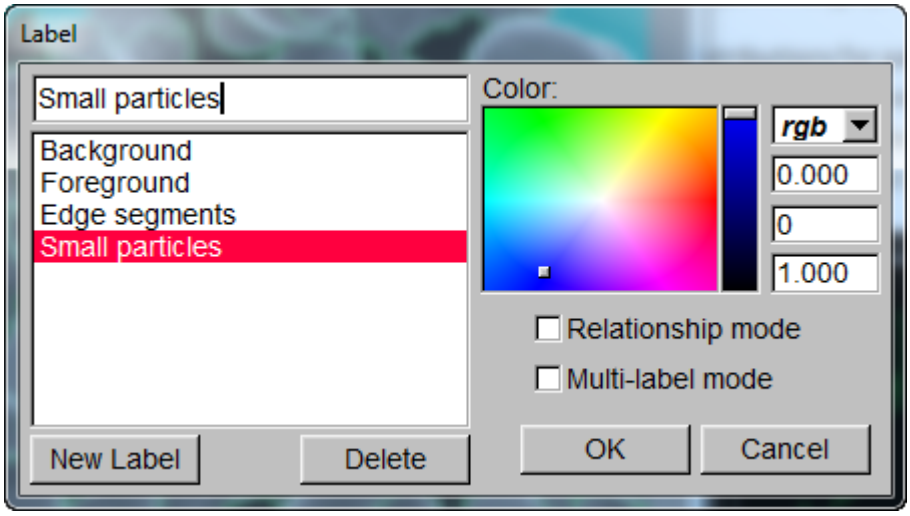
2. Select small particles by selecting the left side of the histogram.

3. Press **Apply** to select the particles on the main image window.

4. Press **Right-click -> Label** in the main image window.

5. Press **New Label** on the Label widget as shown below.

6. Enter a name “Small particles” and select a color for the set as shown below.




The last two histograms viewed can also be plotted as a scatter plot. Select Plots at the top of the Particle Analysis window, and then select scatter plot. The last two histograms viewed will be displayed as a 2D scatter plot. You can display at log or linear scale with! & / buttons. Clicking in the histogram window will allow you to select a set of particles, which you can see highlighted red in the image window. Click apply and these will remain selected for labeling or further segmenting.

Note that if the set of particles that you can select from these histograms is the opposite of what you want, highlight the particles in the histogram tool/2D plot tool, collect apply, then on the image tool bar, click select->inverse set. The opposite set of segments will be selected.

Note: Any colors can be selected, and names can be used, but avoid symbols.

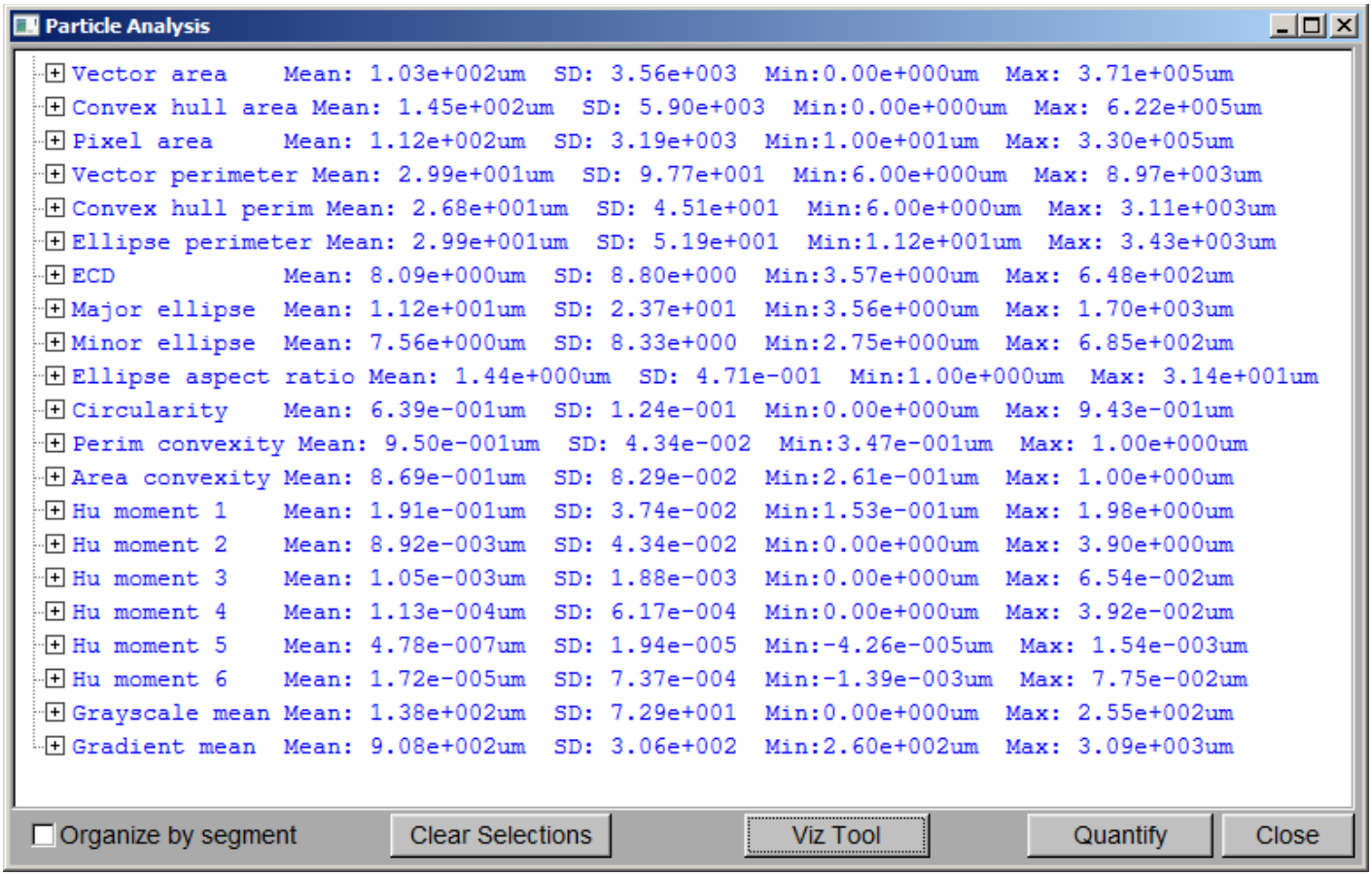
**Development Note:** The “relationship mode” and Multi-label mode are features in development for dealing with more complex structured relationships between features. These should not be checked at this point.

### 1.9.2 Quantification Refinement and Multi-Group Export

Now that a subset of particles has been assigned a new label, you probably want to recalculate the particle statistics. Specifically, we have assigned a label **Small particles** to a subset of the

**unlabeled** particles. Press **Calculate** and now choose **Small particles** from the **Choose Sets** dialog box discussed at the start of this Section.

NOTE: the attributes displayed in the **Particle Analysis** window are the latest group calculated. Go to the **History** menu item to see a list of particle calculations that have been performed in the workflow. If you select one of the older calculations the **Particle Analysis** window will be updated with that analysis. The analysis currently displayed is the analysis that will be exported when you choose that menu option, and will be the one sent to the multi-sample analysis viz tool.



### 1.9.3 Segmentation Refinement of using the quantification results

The interactive labeling tools can be used to clean up segmentation results. In complex images, it is sometime hard to see exactly what set of segments (particles) are being quantified. Reviewing the quantification and looking at the min and max for different attributes can indicate that there are segments that should not be quantified, or that should be merged into a larger particle. We strongly suggest that you review the quantification results, see what the histogram of the particle distribution shows, and then select particles that do not seem to belong and correct the segmentation or eliminate them from quantification (i.e. redo the quantification after re-labeling or re-segmenting sub regions.) For example, looking at the Aspect Ratio calculations can show that two or more particles are merged into an odd shape, and should be split. This is not always apparent in the image on the screen when particles are touching. Another common case is small regions of background between particles that were not eliminated from calculations during labeling—these often have a color, shape, or size difference from the particles of interest. A few iterations of reviewing the attribute calculations and then selecting and eliminating/correcting segments, and recalculating will greatly improve the accuracy of quantification results. The quantification can only be as accurate as the segmentation.

If numerous corrections to the segmentation are needed, it is sometimes easier to select the particles of interest that are correctly segmented and give them a new label, while continuing to iterate on the unlabeled segments. For example, in an image with many overlapping particles, a user may want to select particles that are above a certain size, or within certain shape range, and label then as 'good'. Re-calculating on the remaining unlabeled particles and iterating between the image and calculations can help the user move more non-overlapped particles to the 'good' category. When the user is satisfied that all the particles that can be accurately quantified are labeled, the 'good' particles can be selected for quantification.

Note: For the development of strict quantification protocols where a conduct of ops are needed, image requirements would have to be set that would minimize particle overlaps, particle crowding, and other sources of quantification variability and error.

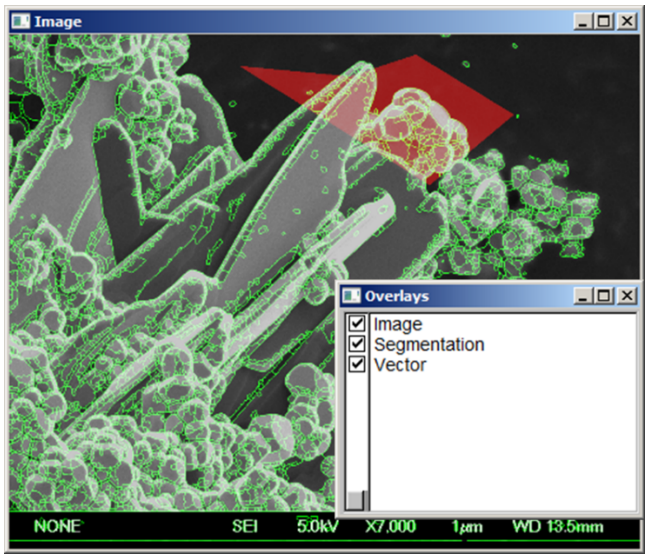
## 1.10 Additional Features

### 1.10.1 The Layers Window

This functionality is in development and provided for advanced users only. Choosing:

**Windows -> Show Layers**

From the main menu will open the **Layers Dialog** as shown below.




There are four main layer types and the image above shows examples of all four.

**1. Image**: This is the original image (or filtered versions of it) displayed in grayscale above. Although you may have filtered images that you use for segmentation, most often you will want to unclick these images and click the first image for use in analysis. There are no checks in place to stop you from performing calculations on altered images.

**2. Segmentation:** This is the particle segmentation, delineated with green lines.

**3. Labeling:** These are labels (or colors) assigned to the segmentation e.g. the aqua edge segments.

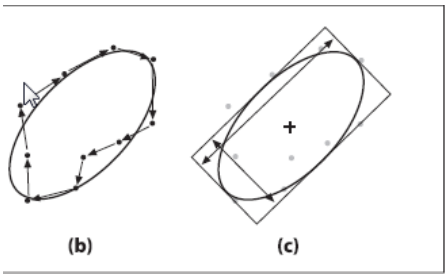
**4. Vector:** Refers to annotations such as the polygon, line, rectangle and paintbrush tools.

Using the checkboxes on the Overlay Dialog, a user can turn off these layers as required for display. However many of the analysis routines require layers to be active. That is, if you uncheck the segmentation layer, you will not be able to run Particle Analysis, and Mama will tell you that it cannot find an active segmentation. Multiple layers of the same type can be present, however it is recommended that only one layer of each type be active at any point in time.

# APPENDIX A: Quantification Attributes:

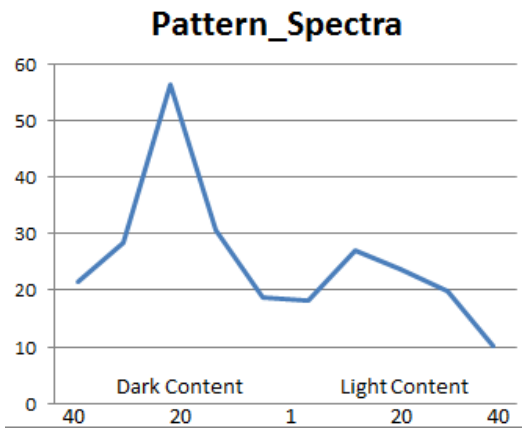
*NOTE: THIS GLOSSARY IS CURRENTLY BEING UPDATED. Some calculations have changed slightly since the glossary was written. We will also provide some calculation limits for selected calculations (i.e. the smallest size object for which you can quantify the attribute with reasonable accuracy.) If you have any questions on exact details of how items are calculated, please ask!*

| Term | Definition |
|---|---|
| **Area and Perimeter**<br><br>**Note: Odd or irregular shapes may give slightly different results than smoothly contoured shapes of the same approximate size** | |
| **Pixel count** | This is the area of the object, not including holes. This is calculated by counting the total number of pixels within the object boundary, and multiplying by the micrometer/pixel scale. For rounded objects, the area calculated this way would always be a tiny amount smaller than the theoretical geometric equivalent because a finite number of pixels are used to represent a curve. |
| **Convex Hull Area** | This is the area of the convex hull (convex polygon outline) of the object. This is calculated using functionality that comes with the OpenCV software: cvFindContours to find the external boundaries of the object and calculates a list of vertices, and does the same for the external boundaries/vertices of the holes. These vertices are used in cvConvexHull2 using Sklansky's algorithm to determine the convex hull. The area of the convex hall is calculated using cvContourArea. This function calculates the region area within the contour consisting of *n* points $p_i = (x_i,y_i)$, $0<i<n$, $p_0 = p_n$, as a spatial moment:<br><br>$$\text{area} = \frac{1}{2}\sum_{i=1}^{n} x_{i-1}y_i - x_i y_{i-1}.$$<br><br>Note: If the input contour has points of self-intersection, the region area within the contour may be calculated incorrectly. |
| **Area** | This is calculated as above, but uses the contour representation of each object, instead of the convex hull, when estimating area with cvContourArea. This calculation does not remove holes before the area calculation is performed. NOTE: this area estimate is typically less than the **Area** estimate based on pixel count, even though the pixel count does not include holes. This is because the cvFindContours approximation of the object boundary passes through the perimeter pixels, reducing the area by a fraction of a pixel all the way around. |
| **Perimeter** | Perimeter is calculated using the object contour determined by cvFindContours. The function cvArcLength calculates the length of the perimeter as the sum of segments between subsequent points. Because we are representing objects that appear round with a finite set of pixels (points) for the boundary, and calculating the boundary as steps/diagonals around the perimeter, this perimeter will typically be slightly larger (a few percent) than the theoretical perimeter of a curved shape. |
| **Convex Hull Perimeter** | This is the perimeter of the convex hull of the object. This is calculated using cvArcLength only using the vertices found using cvConvexHull2. Again, the perimeter is the sum of line segments between subsequent points in the boundary. Even for a convex object, because fewer points are used, this perimeter will be smaller than the pixel based perimeter and often slightly smaller than the theoretical perimeter (because it is estimating a line instead of a curve between points. |

| | |
|---|---|
| | |
| **Ellipse Perimeter** | The perimeter of the best-fit ellipse is estimated using the values obtained from the calculations of the major and minor axes of the best-fit ellipse, with the following equation (where *C* is the estimated perimeter of the ellipse):<br>*a* = length of major radius of ellipse (i.e. half the length of the major axis)<br>*b* = length of minor radius of ellipse (i.e. half the length of the minor axis)<br>$C \approx \pi\,(a+b)\left(1+\frac{3\left(\frac{a-b}{a+b}\right)^2}{10+\sqrt{4-3\left(\frac{a-b}{a+b}\right)^2}}\right);$<br>The equivalent ellipse is calculated using cvFitEllipse2:<br>The cvFitEllipse2() uses a fitting function and returns the ellipse that is the best approximation to the contour. This means that not all points in the contour will be enclosed in the ellipse returned by cvFitEllipse2(). The fitting is done using a least-squares fitness function. The results of the fit are returned in a CvBox2D structure. The indicated box exactly encloses the ellipse. From this the major and minor axis of the ellipse are easily extracted. See Figure below (From OReilly Learning OpenCV : www.cse.iitk.ac.in/users/users/vision/dipakmj/papers/**OReilly**%20Learning%20**OpenCV**.pdf<br>*This means that the ellipse is found that minimizes the geometric distance of the boundary points on the particle from the fitted ellipse. The ellipse is not constrained to be the same area , centroid, or any actual diameter as in the particle. This would minimize odd narrow tails, fringes, or extensions on the particle—in essence; the ellipse is a smoothed version of the particle. ( detailed math on the fitting can be found at* www.cs.unc.edu/Research/stc/FAQs/**OpenCV**/**OpenCV**Reference**Manual**.pdf<br>  <br>(b) (c)<br>Equivalent ellipse area could also be implemented, but is not commonly used. |
| **Axis Diameters**<br>**Note: Odd or irregular shapes may give slightly different results than smoothly contoured shapes of the same approximate size** | |
| **Equivalent Circle Diameter** | This is the diameter of the equivalent circle (diameter of a circle with an equivalent area). Area used is pixel based area, and diameter = $(4A/\pi)^{1/2}$ |
| **Major & Minor Ellipse** | These diameters are calculated from the equivalent ellipsoid (same ellipse as calculated in perimeter) |
| **Aspect Ratios**<br>**Note: Odd or irregular shapes may give slightly different results than smoothly** | |

| | |
|---|---|
| **contoured shapes of the same approximate size** | |
| **Ellipse Aspect Ratio** | This is the ratio of the Major Axis Length to the Minor Axis Length. (1 = sphere, > 1 longer than sphere. |
| **Maximum Chordal Diameter Aspect Ratio** | Firstly, the external boundary of the object is calculated, as described previously. This provides a list of points in the objects boundary. Next, the distance between all pairs of points in the boundary is calculated, along with the angle that the chord between the two points makes with the image horizontal axis. The pair of points that has the greatest distance between them is noted (i.e. the maximum-distance chord), together with this distance measure (chord length) and the angle that this maximum-distance chord makes with the axis. The list of point pairs is then searched to find the pair that has the largest distance between them, but with the constraint that they must be within a small angle (angle tolerance is 5 degrees) of being orthogonal to the maximum-distance chord (the longest orthogonal chord). The maximum diameter aspect ratio is then calculated as the ratio of the length of the maximum-distance chord to the distance of the longest orthogonal chord. Note that neither of these are constrained by going through the centroid; an equilateral triangle would have an aspect ratio of 1 by this calculation. |
| **Shape**<br><br>**Note: Odd or irregular shapes may give slightly different results than smoothly contoured shapes of the same approximate size** | |
| **Circularity** | This is calculated from 4 π Area /(Perimeter$^2$ ) using the pixel based area and perimeter. This is the inverse of the object roundness, and can be useful since it must be between 0 and 1, where circle =1) |
| **Roundness** | This is calculated from (Perimeter$^2$ ) / 4 πArea using pixel based area and perimeter. This will be greater than 1 if not a circle, and 1 for a perfect circle. (The square root of this sometimes called Heywood circularity factor) |
| **Perimeter Convexity** | This is the ratio of the convex hull perimeter to the object pixel perimeter. (Must be between 0-1) |
| **Area Convexity or Compactness** | This is the ratio of the area of the object to the convex hull area (Must be between 0-1) |
| **Invariant Moments**<br><br>**Note: Odd or irregular shapes may give slightly different results than smoothly contoured shapes of the same approximate size** | |
| **Hu 1 to 6** | In image processing an image moment is a weighted average (moment) of an images pixels' intensities by a selected function. The image used is a binary mask of a segmented section of the original image. (i.e. a black/white or 0/1 box around a segmented image section; in this case, a black box with a white particle inside it ), so only contains information about the image (particle) shape, size, and orientation. The moments are calculated using openCVs function cvMoments, which calculates spatial and central moments up to the third order. Moments can be used to calculate the center of gravity, area, main axes, inertia, and other shape characteristics. In this case, the moments are used to calculate Hu moments (cvGetHuMoments). Hu moments are linear combinations of the normalized central moments which are invariant under translation, changes in scale, and rotation. Hu moments are used in image processing for image/shape matching (such as letter recognition), since they are invariant to scale/rotation/translation. 6 Hu moments are calculated, but only 2 are currently displayed. |
| **Simple Texture Measures** | |

| **Note: Odd or irregular shapes may give slightly different results than smoothly contoured shapes of the same approximate size** | |
|---|---|
| **Grayscale mean** | When calculating grayscale statistics for an object, a mask which defines the set of pixel locations belonging to an object is used. At every pixel location defined by this mask, the corresponding grayscale pixel intensity value is extracted and used in calculations.<br><br>Mean (m) is the sum of the pixel intensity values divided by the number of pixels in the particle.<br><br>$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i.$$ |
| **Grayscale variance** | Grayscale variance ($s^2$) is the mean of the squared deviation (difference) of the pixel values from the grayscale mean. (this is the square of standard deviation)<br><br>$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i - \mu)^2.$$ |
| **Gradient Mean** | This uses cvSobel (Sobel operator, a differentiation operator) to calculate an approximation of the gradient of the image pixel intensity values in the x and y directions, and then combines them to give a single gradient magnitude value for each pixel location. The mean of these values are then calculated. |
| **Gradient Variance** | This is the variance in the gradient magnitude values. |
| | |
| **Texture quantification**<br><br>**Note: Odd or irregular shapes may give slightly different results than smoothly contoured shapes of the same approximate size** | |
| | |
| **Pattern Spectra / Granulometry** | A pattern spectrum (or granulometry) is a technique from mathematical morphology for evaluating the distribution of sizes within images. An image is filtered at a number of different scales and the effect of filtering at each size is measured. The resulting set of numbers describes the distribution of features in the image across different scales and has been shown to be useful in characterizing textures. |

| | |
|---|---|
| | Mama implements filtering based on *opening* and *closing*, and uses circular structuring elements. The filtering scale is user defined by specifying a minimum and maximum scale, as well as the number of different scales (or step size).<br><br>The pattern spectra produces two values at each scale: one associated with *opening* and one associated with *closing*, however these are typically treated as a single plot as shown in the figure. For example, assuming the minimum scale was 1 pixel and the maximum scale was 40 pixels, the pattern spectra begins on the left at 40 pixels with *closing*. As the plot moves to the right, the scale grows smaller, until it reaches 1 pixel. The plot then increase from 1 pixel through to 40 pixels with *opening*. At a high-level, a user can interpret the plot as the amount of dark and light content in the image at the particular scale.<br><br>Plotting the pattern spectra as a function of scale is useful for interpretation, however, for automated processing, it is often useful to reduce the number of values. Mama calculates the statistical moments of the pattern spectra: the mean, variance, skewness and kurtosis.<br><br>  |
| **Grey Level Co-occurrence (Entropy, Energy)** | Possibly the most commonly used method for texture analysis in the image processing literature is the Gray-Level Co-occurrence Matrix (GLCM), first described by Haralick[1]. This approach is based on the use of second-order statistics of the gray-level histogram.<br><br>The Gray Level Co-occurrence Matrix, GLCM is a tabulation of how often different combinations of pixel brightness values (gray levels) occur in an image. The GLCM is used for a series of "second order" texture calculations. These measures consider the relationship between groups of two pixels, in a particular spatial relationship, in an image.<br>Mama uses the ITK implementation of GLCM with the default spatial offset (1, -1). Once the co-occurrence matrices are calculated from the image data, for the specified spatial relationships, various features can be calculated. The ITK implementation of the cooccurrence features calculates the 6 features recommended by Conners, Trivedi and Harlow[2]:<br><br>$\text{"Energy"} = f_2 = -\sum_{i,j} g(i,j) \log_2 g(i,j)$<br><br>$\text{"Entropy"}\ g(i,j) = 0, \text{ or 0 if} = f_3 = \sum_{i,j} \frac{(i-\mu)(j-\mu)g(i,j)}{\sigma^2}$<br><br>$\text{"Difference Moment"} = f_5 = \sum_{i,j} (i-j)^2 g(i,j)$<br><br>$\text{"Inertia"} = f_6 = \sum_{i,j} ((i-\mu)+(j-\mu))^3 g(i,j)$ (sometimes called "contrast.")<br><br>$\text{"Cluster Shade"} = f_7 = \sum_{i,j} ((i-\mu)+(j-\mu))^4 g(i,j)$<br><br>$\text{"Cluster Prominence"} = f_8 = \frac{\sum_{i,j} (i,j) g(i,j) - \mu_t^2}{\sigma_t^2}$<br><br>The Mama implementation calculates the 6 GLCM features at various scales. As with the Pattern Spectra, these scales are user defined by by specifying a minimum and maximum scale, as well as the number of different scales (or step size). The mean and variance of each feature is also |

| | |
|---|---|
| | calculated to provide a scale invariant description.<br><br>[1]Haralick, R.M. 1979. Statistical and Structural Approaches to Texture. Proceedings of the IEEE, 67:786-804. [2]R.W. Conners, M.M. Trivedi, and C.A. Harlow. Segmentation of a High-Resolution Urban Scene using Texture Operators. Computer Vision, Graphics and Image Processing, 25:273-310, 1984. |
| **Additional measures partially implemented but not included in v1.2 release**<br><br>**Note: Odd or irregular shapes may give slightly different results than smoothly contoured shapes of the same approximate size** | |
| | |
| **Fractal Dimension/Fractal analysis** | P. Soille and J.-F. Rivest (1996). "On the validity of fractal dimension measurements in image analysis". Journal of Visual Communication and Image Representation **7**: 217–229. doi:10.1006/jvci.1996.0020. http://mdigest.jrc.ec.europa.eu/soille/soille-rivest96.pdf. & http://products.asminternational.org/hbk/index.jsp, Volume 7, Powder Metal Technologies and Applications -> Metal Powder Production and Characterization -> Particle Image Analysis -> Particle Shape Characterization. This has not yet been implemented. |
| **Fourier analysis (geometric signature waveform** | Fourier transform of the centroid to normalized perimeter 'radius' calculated at 1 (or 0.1 degree angles) and plotted as a 2 D wave form plot. http://products.asminternational.org/hbk/index.jsp, Volume 7, Powder Metal Technologies and Applications -> Metal Powder Production and Characterization -> Particle Image Analysis -> Particle Shape Characterization. This has not yet been implemented. |
| **Entropy** | Entropy is a measure of the uncertainty associated with a random variable. It can be thought of as a statistical measure of randomness, and in image processing applications it is often described as a measure of the amount of information contained in an image. The estimation of an object's entropy is calculated from the histogram of intensity values in the object. The Entropy, of an object, $X$, having $n$ possible values, $H(X)$, is defined as:<br><br>$$H(X) = \sum_{i=1}^{n} p(x_i) \log_2 p(x_i)$$<br><br>Where $p(x_i)$ is the probability of occurrence of a pixel grayscale value of $x_i$. These probability values are calculated from the grayscale histogram of the object. Low entropy images have little variation and large runs of pixels with the same or similar grayscale values. An image that is perfectly flat will have an entropy value of zero. On the other hand, high entropy images have a great deal of variation in the values of the pixels in the image.<br><br>*(currently being implemented)* |
| **Statistical Moments (3rd): Skewness** | This is a measure of the symmetry (or lack of symmetry) in the distribution of intensity values of the pixels in the object. The pixel distribution is symmetric (skewness= 0) if the distribution is the same on both sides of the center point. Positive or negative values indicate that the distribution is skewed in one direction or the other (negative or left skew indicates a tail on the left side of the distribution is longer & the mass is concentrated on the right with relatively few lower values) Skewness can be calculated using standard formulas for grayscale pixel intensity values, and for gradient magnitude values.<br><br>*(currently being implemented)* |

| | |
|---|---|
| **Statistical Moments (4th): Kurtosis** | Kurtosis is a measure of whether the data (intensity of pixel values) are peaked or flat relative to a normal distribution. Distributions with a high kurtosis have a distinct peak near the mean and decline rapidly. Pixel intensity distributions with low kurtosis have a flat top near the mean rather than a sharp peak. A standard normal distribution has a kurtosis of 3; therefore, *kurtosis* reported here is 'excess kurtosis' or kurtosis minus three, giving a standard normal distribution a *kurtosis* of 0 by this definition.<br><br>*(currently being implemented)* |
| **Heterogeneity** | This is the fraction of pixels in the object that deviate more than a set range (X%) from the object's mean intensity. (Eventually this will be set with a user defined variable X, but currently implemented with X = 10%. |
| **Weibull Shape and Scale** | The weibull distribution is a continuous probability distribution. The probability density function of a Weibull random variable x is<br><br>$$f(x;\alpha,\beta) = \begin{cases} \frac{\beta}{\alpha}\left(\frac{x}{\alpha}\right)^{\beta-1} e^{-\left(\frac{x-\mu}{\alpha}\right)^{\beta}} & x \geq 0, \\ 0 & x < 0 \end{cases}$$<br><br>Where $\beta > 0$ is the shape parameter, and $\alpha > 0$ is the scale parameter of the distribution, and $\mu$ is the origin of the distribution.<br><br>In our case, for an image of a segmented particle, image (pixel intensity) gradients and the distribution of their magnitude are calculated. First order directional Gaussian derivative filters Gx; Gy are applied to image I:<br><br>$$G_1 = \frac{dG(x,y)}{dx} \; ; G_2 = \frac{dG(x,y)}{dy} \; ; G(x,y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$<br><br>in order to compute the gradient magnitude:<br><br>$$|\nabla I(x,y)| = \sqrt{[I(x,y) \otimes G_1(x,y)]^2 + [I(x,y) \otimes G_2(x,y)]^2}$$<br><br>A Weibull fit for the distribution of gradient magnitudes is determined, in order to obtain the scale and shape parameters. Specifically, In the equation above, x > 0 is then the edge response of a Gaussian derivative filter, and $\mu$ is the origin of the contrast distribution. $\mu$ is eliminated by stretching the contrast to full range. This gives us a shape and scale parameter for each particle in the image, as a representation of its overall texture. |
| **Gabor Texture Features: Gabor Filter& Wavelets** | These are Orientation and scale tunable edge / line detectors. Gabor filters are a commonly-used method for texture analysis. Their popularity is due in part to the fact that they seem to have similar functionality to certain parts of the early stages of the mammalian vision system. In the spatial domain, a 2-D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave. The various parameters of the filter define specific orientations and sizes to which it will be sensitive. However, it is this sensitivity to scales and orientations that can be problematic in various applications. Often objects of interest are observed under various orientations, and therefore conventional Gabor representations yield unacceptable performance. There are various freely-available implementations of Gabor filters. However, we have found none that have a rotationally-invariant implementation. Thus, we need to modify the existing code to make them rotationally invariant. There are methods described in the literature for doing this; |

| | |
|---|---|
| | however we have yet to fully implement this texture option. |
| **Clumpiness** | This fraction of pixels deviating from the average remaining after dilation, reflecting texture variation. This is a variation on heterogeneity, requiring two variables (dilation parameter and deviation percentage) and will be fully implemented when the user-defined heterogeneity parameter is implemented |
| **Margination** | This relative distribution of the object intensity between the center and the margin (large value =brighter center). This required multiple parameters, with no existing functionality in software packages. This will be implemented at a later date. |

# Appendix B: Data Storage

Images opened in MAMA are copied and stored in the /Program Files/Mama/dataV2/ImageFiles directory for Windows systems. Image files are copies of the image file originally opened in MAMA and are never modified. Workflows associated with an image are stored in the /Program Files/Mama/dataV2/WorkFlowFiles directory. Workflows are named with a prefix beginning Imagewf and then a workflow number (Example: Imagewf3we60.fits where 3 is the workflow). Workflow files belonging to the same workflow will have identical workflow identifiers (in this example, wf3). There is no way to directly identify from the file name which image the workflow is associated with. Each step in the Workflow Window that creates a change in the image is stored as a separate file in the workflow directory. Be aware this can result in the creation of a very large number of workflow files associated with each image and these files can be quite large. NEVER DELETE WORKFLOW FILES DIRECTLY FROM THE WORKFLOWS DIRECTORY. Doing so will cause image/workflow corruption and the possible inability to open images in MAMA. To delete workflows from MAMA, select the workflow you want to delete (Workflow Summary->File>Delete Workflow). You can also delete an image and all workflows associated with that image (Workflow Summary->File->Delete Image). You will be prompted to confirm you want to delete.

# Appendix C: Credits and Contacts

Questions about obtaining the Mama software, and / or project history, management and the nuclear forensics application domain please send email to:

mama-info@lanl.gov

For technical questions regarding the software, image processing routines, bugs, issues or comments, please send email to the technical software mailing list:

mama@lanl.gov

The MAMA software is currently developed and supported by the MAMA Team directed by Jeffery Bloch (jbloch@lanl.gov) at Los Alamos National Laboratory. The software was originally designed by Reid Porter and Christy Ruggerio at Los Alamos National Laboratory.



We gratefully acknowledge the support of the U.S. Department of Homeland Security through the Domestic Nuclear Detection Office for this work.