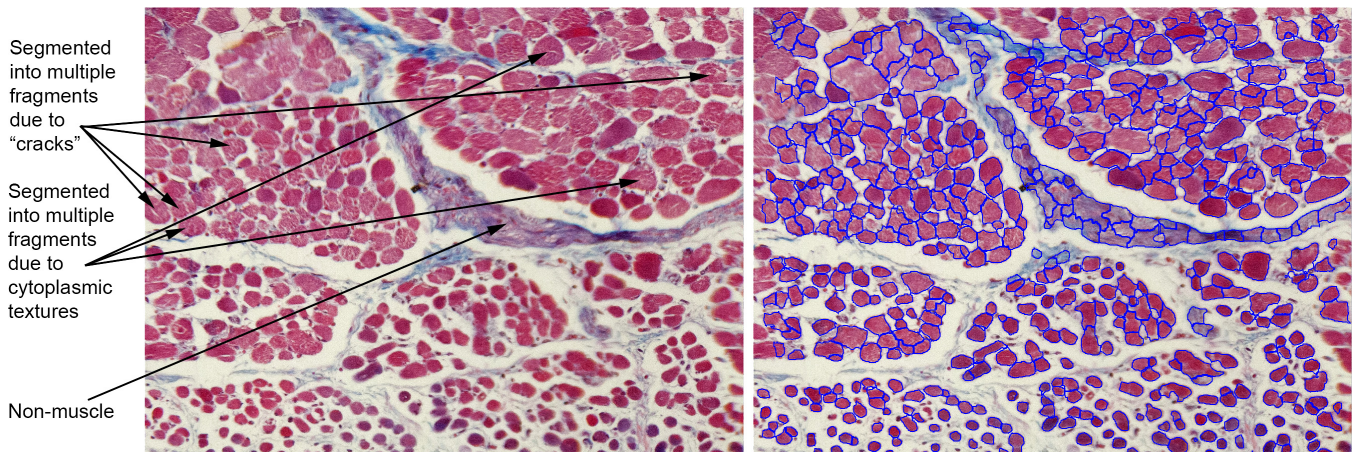


Photographing & Segmenting Muscle Fibers in EOM Crossections

JM Miller, PhD
Eidactics, San Francisco

Quantitative histology of extraocular muscles (EOMs) conventionally uses sampling to avoid having to count, or worse to measure, the thousands of fibers in each crosssectional slice, but unfortunately this tissue is highly inhomogeneous, and unbiased sampling is difficult. We wish to circumvent the sampling problem by segmenting essentially all the muscle fibers in stained EOM slices and calculating their areas. But it is impractical to delineate tens or hundreds of thousands of fibers by hand, and yet a completely automated process appears infeasible.

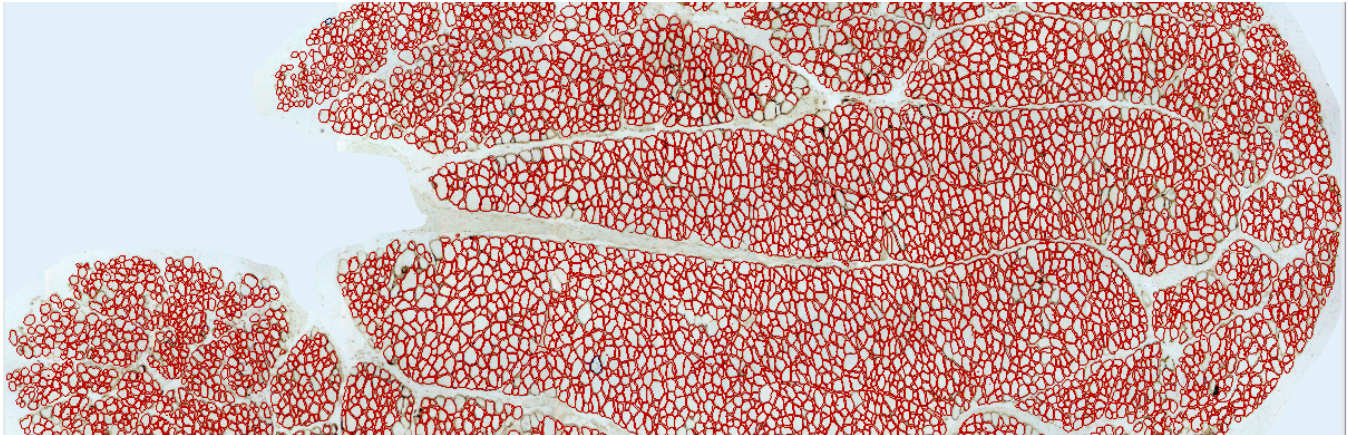
Stained cell bodies appear in crosssection as small, tightly-packed roundish objects. Formalin-fixed, paraffin-embedded (FFPE) fibers shrink, creating surrounding voids that make segmentation easier, but distort size measurements. Flash-freezing (Cryo-processing) preserves fiber sizes, but algorithmic segmentation is difficult: cell boundaries must be detected as step changes in luminance between the stained cell body and surround, which leads to many spurious boundary detections, and nonuniform staining of cell bodies causes cell fragments to be wrongly segmented as cells, both of which must be corrected manually.



Poor segmentation of stained cell bodies.

A better approach is to stain fiber boundaries, eg, with an immunohistochemical stain for dystrophin, a protein associated with the sarcolemma. Cell boundaries can then be detected as dark lines, instead of step changes in luminance, reducing spurious detections. The problem of nonuniform staining of cell bodies is minimized. Non-muscle structures are not stained. The burden of manual correction is greatly reduced. Our general procedure is:

- Flash-freeze muscle samples and stain for dystrophin, maximizing sarcolemmal staining, and minimizing cell body and other background staining.
- Mask or paint-out non-muscular regions that would confuse segmentation algorithms.
- Segment tightly-packed fibers using a Watershed algorithm.
- Remove incorrectly segmented fibers by color (eg, cell bodies must be light-colored) and shape-based (non-roundish putative cells are rejected) filtering, where thresholds can be set interactively while assessing their overall effect on the image.
- Smooth contours and fix residual segmentation errors (fiber fragments, groups, non-fibers and missing fibers) manually.
- Calculate fiber areas to a "live" spreadsheet, where suspect entries can be selected to view corresponding image regions for possible editing. "Roundness" indices, and variance measures help find segmentation errors.



Good segmentation of stained cell outlines.

Systems, Software & Files

- (1) Each tissue section is photographed as a set of overlapping fields with Leica LAS on a dedicated Windows 7 Professional machine.
- (2) Each set is assembled into a montage with the Photoshop PhotoMerge function. Photoshop is also used to clean-up images and optimize contrast.
- (3) The montage is segmented and quantified with IP-10 on Windows 10, running under MacOS BootCamp.
- (4) All files are stored under MacOS on "Rust", using the descriptive, hierarchical file structure, described below. All other file versions are considered temporary and should be deleted to avoid confusion.

There are several issues in moving files between MacOS and Windows:

- iCloud updating delays – although it is possible to access iCloud from Windows running under BootCamp as well as from macOS, updating of our large image files sometimes takes several days!
- Different interpretations of filename characters by MacOS and Windows.
- Windows & IP-10 limits on pathname length, eg, IP-10 will not open a file with a long pathname, though it gives no error message!

Thus, we move files between MacOS and Windows on the top level of an external drive, storing them in the descriptive MacOS hierarchy on "Rust" as the final step.

Histological Processing

- Muscles are flash-frozen, cryo-sectioned, and stained for dystrophin, as described elsewhere.
- If sarcolemmal staining is not strong and uniform, with the cytosol remaining unstained, and the slide is not free of debris, algorithmic segmentation will be poor, and a great deal of manual work may be required!
- Each section is mounted with long axis parallel to one or the other edge of the slide to facilitate scanning.
- Slides must be labeled to completely and unambiguously identify them, as follows:

- Sample (eg: **R-4468-LIO-s2-d2b** means:

- Rabbit ID 4468
- Left Inferior Oblique muscle
- Slice ~ 2 mm from the origin
- sliDe #2, section b on that slide.

Use upper and lower case, and punctuation exactly as shown.

- Fixation & Embedding (eg: **Cryo-20180928**, where 20180928 is the date of flash-freezing with case & punctuation exactly as shown.
- Staining (eg: **Dys-20190125**, where 20190125 is the staining date).

a	R-4468-LIO-s2-d1
b	Cryo-20180928
c	Dys2-20190723
d	Eidactics • SRF

Every slide that will be retained must be clearly labeled with this information. A "PTouch" template is available to create such labels and should be pasted over the penciled labels used during processing, eg:

Photographing Stained Slides

Cleaning Slides

- (1) Blow slide clean with compressed air.
- (2) Apply a drop of lens cleaning solution to each side.
- (3) Gently wipe with a fresh piece of lens cleaning paper.

Leica DM6000-B Microscope and LAS Software Set-up

- (1) Be sure microscope optics are clean.
- (2) Be sure the slide is squarely positioned and flat on the stage.
- (3) Select the 10x objective.
- (4) Set illumination power so that eyepiece image has optimal resolution.
- (5) Adjust the Field Diaphragm if necessary for proper illumination.
- (6) Adjust the Aperture Diaphragm for the sharpest image without diffraction artifacts (Köhler Illumination¹).
- (7) Adjust focus knobs until the slide is clear and sharp in the camera image.
- (8) » **LAS » Acquire » Camera » DFC425 C - Last Used » Automatic white balance on whole image** = Click
- (9) » **LAS » Acquire » Camera » DFC425 C - Last Used » Automatic exposure** = Off
- (10) Photograph the Calibration Slide under the same conditions.

Slide Scanning

Take successive photos in a systematic way across the sample, spaced so that overlap of adjacent frames is not excessive, but all parts of the sample are captured. Excessive overlap will cause Photoshop's PhotoMerge function to fail (20-30% linear overlap is usually good).

Processing CRAW Slide Images into a Montage Using Photoshop

PhotoMerge

- (a) » **File » Automate » PhotoMerge....**
- (b) **Layout: Reposition.** This translates and rotates, but does not distort component images.
- (c) **Use: Files**
- (d) **Browse** to select the cropped images to be merged.
- (e) Select only **Blend Images Together**
- (f) **OK** to run. PhotoMerge may require several hours to complete. It may also freeze at the end of "Align Selected Layers Based On Content" (progress ~20%). Reducing number of images and image overlap may solve this problem.
- (g) Image names continue from the scheme above, each beginning, eg, with **R-4468-LIO-s2-d2b Cryo-20180928 Dys-20190125.**
- (h) Save the multilayer montage, eg, as: **R-4468-LIO-s2-d2b Cryo-20180928 Dys-20190125 - Layers.psb.**
- (i) Flatten the image.

Optimize Color Contrast

- (a) In the flattened montage, use » **Image » Adjustments » Levels** to optimize the gamut of each of the Red, Green & Blue channels (adjust the left and right Input Level sliders), while keeping the background as close to neutral grey as possible (adjust the center Input Level slider)
- (b) Save as TIFF under a new file name (eg: (eg: **R-4468-LIO-s2-d2b Cryo-20180928 Dys-20190125 - Flat.tif**).

¹ Köhler illumination produces even illumination of the sample while defocusing the image of the illumination source (eg, the lamp filament) so it is not visible.

File Management

- Retain the original photomicrographs in a folder named **LAS Images**. Retain the **multilayer.psd** file and the flattened, color-optimized files. Delete all other, easily regenerated files.
- Make a new folder for each processed slice. Store **LAS Images** folder, the **- Layers.psd** file, and the **- Flat.tif** file within the new folder.

IP-10 Options, Tools & Files

Image-Pro 10 opens, processes, and saves image files. Some IP-10 operations modify the image itself, whereas others create data in “image overlays”, which are included² when the file is saved as TIFF (use ZIF compression, which is non-lossy).

IP-10 Program Options

Set » **File » Image-Pro Options » Application » Display: Individual Object Display Limit** = 20,000.

Set » **File » Image-Pro Options » Application » Memory Management » Memory Limit** = Max.

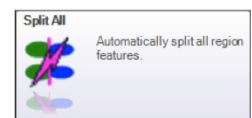
Set » **File » Image-Pro Options » Image » General » Opacity of Inactive Overlays (%)** = 100%.

Set » **File » Image-Pro Options » Image » General » Maximum number of undos** = 50.

The *Muscle Segmentation* Ribbon Contains Most Necessary Tools

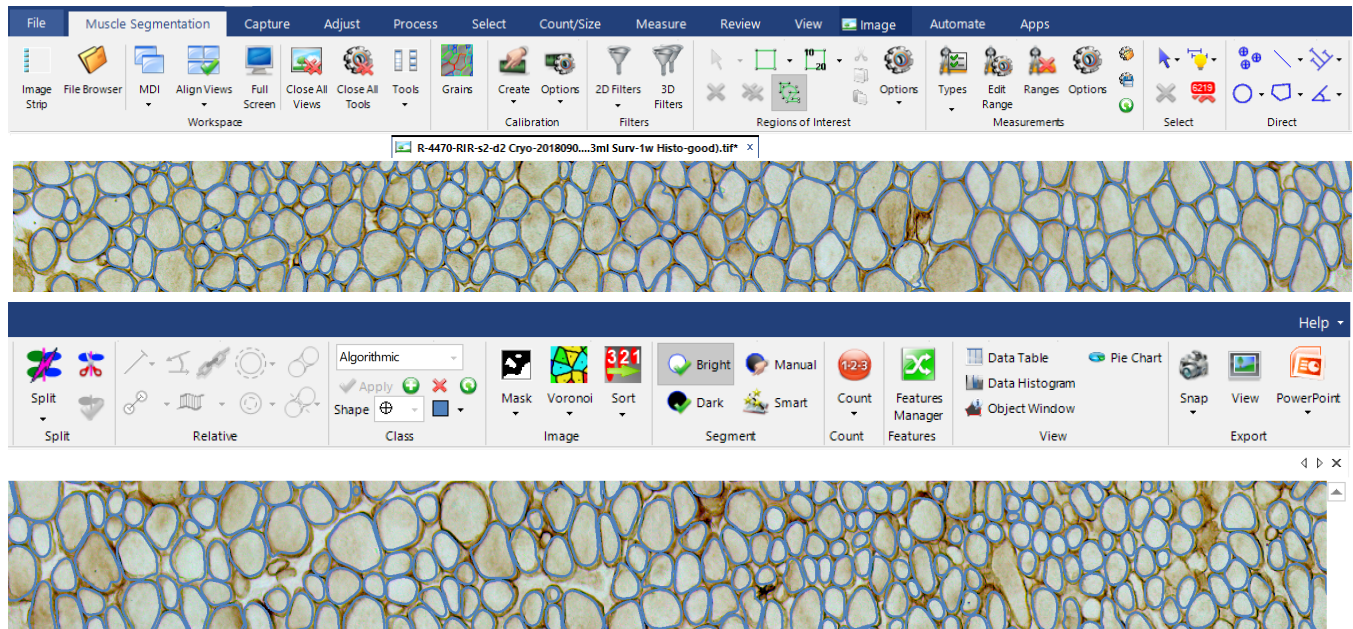
Our workflow uses the tool groups in the custom **Muscle Segmentation** ribbon, roughly left to right:

- » **Workspace » Image Strip** – show or hide thumbnails of open images.
- » **Apps » Grains** – Watershed analysis.
- » **Calibration » Create** – Convert from pixels to μm using included image of calibration slide.
- » **Filters » 2D Filters** – useful for preparing images.
- » **Regions of Interest** – limits the region to which the Grains app, or any other operation, is applied.
- » **Measurements » Types** sets filters applied by the Grains app and the built-in segmentation function after the image is separated into regions, and determines measurements computed.
- » **Measurements » Ranges** toggles the filters. When a fiber fails to be identified, rerun Grains with filters off so see if it was in fact segmented, but was subsequently filtered out.
- » **Measurements » Options** toggles the Measurements Options details panel.
- » **Select** tools select, delete, show, and hide measurement objects.
- » **Direct » Polygon** draws measurement objects that need to be manually added.
- » **Split** draws lines to split groups of fibers wrongly identified as one. Draw multiple split lines and then apply them all with a single click of the **Apply** button. **Do not use the Split All button**, which will automatically process the entire image according to its own inscrutable logic.
- » **Relative » Morphological Close** combines multiple adjacent measurement objects into one. Note that distant objects are not combined, making this an efficient way to combine fragments into whole fibers.
- » **Class** puts measurement objects into multiple named classes for subsequent identification or selective processing. We define various classes to record image processing.
- » **Image » Mask** creates a mask separating measurement objects. The mask is a new binary image that can be filtered and segmented like any image.
- » **Segment & Count** runs the built-in segmentation function. Although the Grains app produces better segmentation of the original image, the built-in segmentation function has a smoothing parameter, which makes it useful in refining segmentation (see below).
- » **Features » Features Manager** – copy and paste measurement objects.



² Overlays are stored as vector objects in TIF tags using an IP-10 format. It is possible to save overlays as separate image files by making the image white and “burning” the overlay in.

- » **View** tools show numeric & graphical results.
- » **Export » Snap** – create an image with measurement objects “burned in” (merged with the image, rather than in a separate layer).



File Management

Several people work on these files over a long period, and file names must be searchable, so it is essential that naming conventions be strictly followed. File names are derived from slide names as in the following examples:

UOIT - Rabbit EOM Dys Fiber Analysis /

R-4470-RIR-s2-d2 Cryo-20180906 Cut-10 Dys-20191002 (Inj-BPX,1.5%,0.3ml Surv-1w Histo-good IP10-done) /

R-4470-RIR-s2-d2 Cryo-20180906 Cut-10 Dys-20191002 (Inj-BPX,1.5%,0.3ml Surv-1w Histo-good) - Layers.psb

R-4470-RIR-s2-d2 Cryo-20180906 Cut-10 Dys-20191002 (Inj-BPX-1.5%,0.3ml Surv-1w Histo-good) - Flat.tif

R-4470-RIR-s2-d2 Cryo-20180906 Cut-10 Dys-20191002 IP10-2020081716 (Inj-BPX-1.5%,0.3ml Surv-1w Histo-good) - Seg1.tif

R-4470-RIR-s2-d2 Cryo-20180906 Cut-10 Dys-20191002 IP10-2020081716 (Inj-BPX-1.5%,0.3ml Surv-1w Histo-good) - Seg2.tif

R-4470-RIR-s2-d2 Cryo-20180906 Cut-10 Dys-20191002 IP10-2020081716 (Inj-BPX-1.5%,0.3ml Surv-1w Histo-good) - Meas.txt

R-4470-RIR-s2-d2 Cryo-20180906 Cut-10 Dys-20191002 IP10-2020081716 (Inj-BPX-1.5%,0.3ml Surv-1w Histo-good) - Burn.tif

The initial part of folder & file names, eg, **R-4470-RIR-s2-d2 Cryo-20180906 Dys-20191002** identifies the sample and processing, and must never vary in any way. Less formal “hints”, suggesting experimental and other conditions, are appended in parentheses, eg:

- **Inj-BPX,0.3%,3ml** – sample was treated with bupivacaine 0.3%, 3 ml.
- **Inj-sal** – control sample, treated with saline.
- **Surv-1w** – rabbit survived 1 week after injection.
- **IP10-20200707jmm** – image was analyzed using Image Pro 10, on 2020-07-07 by jmm.

A suffix is added to distinguish file contents. In the end, the following six files are retained. Any additional files should be placed in an "Obsolete" folder:

- - **Layers**
- - **Flat**
- - **Seg1**
- - **Seg2**

- - Meas
- - Burn

Files that would be difficult to replace, such as the multilayer file, should be made read-only (“locked”) so they are not inadvertently overwritten.

Preparing an Image for Segmentation

Removing Non-muscular Regions

Adobe Photoshop is familiar and easy to use. Choose a soft-edged brush to paint over any vasculature, nerves, or clumps of connective tissues that might be wrongly segmented as muscle.

IP-10 has tools for manual editing and masking objects³, which may be used instead of Photoshop. One way is to use » **Regions of Interest** » **Options** to limit the actions of the Grains app, which begins by optionally masking the image. A mask is a binary image, black for regions to be ignored. It can be created using any or all of the following methods:

- Use » **Segment** » **Smart** » **Smart Segmentation** to select some Objects and Background areas (figure =>). Smart Segmentation will use the selected Recipe Options to segment the image.
- » **Segment** » **Smart Segmentation** » **Mask and Overlay** » **Create Mask**
- **Convert** » **Mono 8bpp** » **Apply**
- **Filters** » **2D Filters** » **Morphological** » **Close** to remove small black areas inside white regions (eg, use the 7 x 7 Octagon structuring element).
- **Filters** » **2D Filters** » **Morphological** » **Dilate** to expand white regions, assuring no muscle is masked out.
- Draw ROIs around regions to be excluded using the » **Regions of Interest** » **Polygon Tool**, and setting them to 0 (**Math** » **Calc** » **Set 0** » **Apply**).

The resulting mask can then be saved and loaded into Grains.

Calibrating the Image

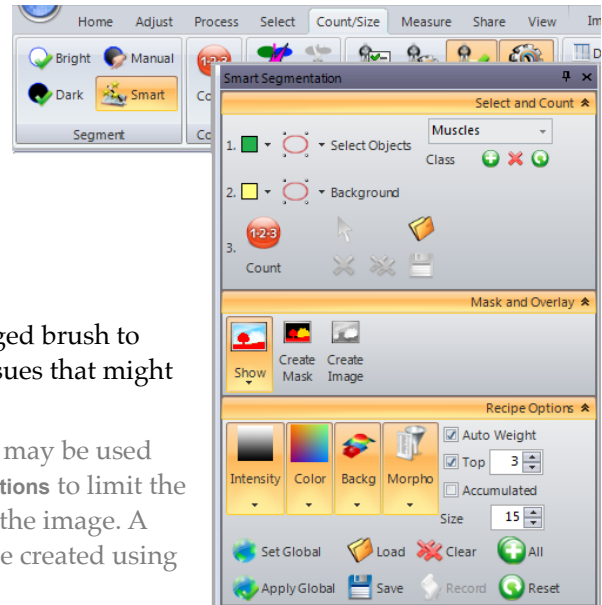
Convert measurements from pixels to μm using an image of a calibration slide created with the same magnification as the muscle images.

- » **Calibration** » **Create** » **Quick Calibration**

Visualize the calibration as recorded in the program, which should match the calibration in the image:

- » **Calibration** » **Options** » **Auto Marker** = ON
- » **Calibration** » **Marker Options** » **Text Size** = 1
- » **Calibration** » **Marker Options** » **Marker Size** = 1
- Click » **View** » **Data Histogram** and select the measure to be shown in the histogram from the drop-down menu.
- Select the measures computed for each segmented object using » **Measurements** » **Types**.

Some of the Measurements Filters are parameterized in terms of the calibrated realworld values. **Grains** values, however, are not.



³ By default, objects are locked following a **Count-Size** operation; use the **Unlock Objects** menu item to unlock all objects.

Segmenting an Image with the *Grains* App

The *Grains* app, **Grains.ipx**, is an add-in module for IP-10, able to segment regions even if their borders are not completely closed. Regions (treated as wholes) are then retained or discarded according to » **Measurements** » **Types**.

If the Grains app panel does not appear in the left sidebar, open it from **Apps**, or if necessary, from **Search** » **Open Project** (both in **Apps** ribbon). It has a few parameters that need to be set (suggested values shown):

- » **Class** = Algorithmic - for non-manual segmentations.
- » **Grains** » **Edge Type** = Dark - for muscle fibers outlined with dark dystrophin staining.
- » **Grains** » **Edge Width** = 15 - small values detect thinner edge boundaries, but are more sensitive to noise.
- » **Grains** » **Sensitivity** = 1 - high values detect low luminance “ridges”, tending to separate fibers into fragments (see “Watershed Separation” below). Values that are too low will fail to separate distinguishable fibers.
- » **Grains** » **Mask** - optionally used to mask-out areas in the image that would be wrongly identified as muscle fibers.
- » **Grains** » **Interactive** - when checked, the Grains app is automatically run whenever a parameter is changed. This is useful for trying parameters within a small ROI.
- » **Grains** » **Measure Grains** - runs the Grains app.

Set » **Measurements** » **Ranges** = ON to enable filtering, or OFF to see the unfiltered result of *Grains* segmentation.

Useful Grains Parameters & Measurements Filter Values

	Dys2 (μm)	Dys (μm)				
Grains » Edge Type	Dark	Dark				
Grains: Edge Width (small values may find spurious contours; large values underestimate fiber areas)	10	15				
Grains: Sensitivity (high sensitivity may find spurious contours)	1	1				
Region: Area (very large & small regions are likely non-fibers)	100 - 6,000	50-5500				
Region: Roundness (non-round regions are likely groups or non-xsections)	1.0 - 3.0	1.0-2.75				
Region: Aspect Ratio (another measure of roundness)	1.0 - 4.5	1.0-3.				
Region: Intensity, Mean (Dys-stained cell bodies are whitish)	100 - 230	100-250				
Object: Class (distinguish Grains, Split, Added & Merged fibers)	(no filter)					
Region: Holes (identifies small, one pixel measurement objects)	(no filter)					

Test Grains Parameters on Part of an Image

Grains works by segmenting the image (see below), and then applying filters to the segments. Grains’ parameters may need to be optimized for each image:

- Use » **Select** » **Square Marquis** to draw an ROI around a typical part of the image, and then **Select** it.
- Use » **Measurements** » **Ranges** to toggle filtering so you can see filtered & unfiltered separations.
- Set » **Measurements** » **Options** » **Segmentation Options** » **Clean Borders** = **None**. Otherwise, this is set to **All**, which discards regions on borders of the image or ROI (if fibers, they would be partial), but for setting Grains parameters, we want nothing discarded.

- Check **Interactive**.

Now, vary **Edge Width** and **Sensitivity** until muscle fibers are well separated. Segmentation errors will subsequently be repaired manually, and your judgment of what constitutes a good Grains separation should be determined by the difficulty doing the various types of repair.

Run the Grains App

- Set the active Class to **Algorithmic** to associate a label and color to each object found by Grains. Classes are distinguished by color in the main window, and by color and Feature Name prefix in the Measurement Table.
- Set » **Measurements** » **Ranges** = ON
- Click » **Grains** » **Measure Grains** to run app.

Show or hide the segmentation with » **Select** » **Lighting**.

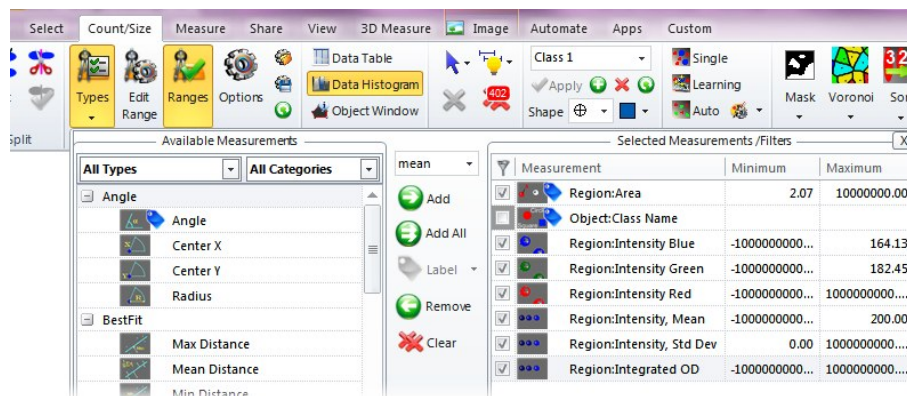
How the Grains App Works

The Grains app performs a series of operations. After applying the **Mask**, if any, edges are extracted and smoothed with 2 filters, both of which use the » **Grains** » **Edge width** parameter to define filter size.

- A Variance Filter extracts edges. This filter associates with each pixel the statistical variance of pixel intensities in its neighborhood. Pixels near grain edges have high variance (due to intensity differences across the edge, plus noise), whereas pixels inside grains and in the background have lower variance (noise only). Smaller filter kernels detect thinner edge boundaries, but are more sensitive to noise.
- A Gaussian Filter then smooths extracted edge contours. Larger kernels increase smoothing.

The image is then separated into regions with a Watershed algorithm, using seed points in uniform areas, between edges found by the Variance Filter. Watershed treats the image as a (greyscale) luminance landscape, as though gradually filling with water to a uniform height. There will be multiple pools at first, but as the water rises pools merge. Every point at which pools merge is on a “ridge”, and the set of these points are “watershed lines”, which are taken to separate objects. Low ridges, which may reflect only noise, are ignored by choosing a “pre-flood” level, below which merging of pools is ignored:

- Watershed Pre-flood ignores Watershed ridge points up to a luminance level determined by » **Grains** » **Sensitivity**.
- The Watershed algorithm then divides the image into regions.



Regions (treated as wholes) are then retained or discarded according to » **Measurements** » **Types** » **Selected**. The **Measurements/ Filters** list actually serves two functions: [1] All items in the list appear as columns in » **View** » **Data Histogram**, and [2] » **Measurements** » **Types** items with a ✓ are used to filter the results of Watershed separation. » **Measurements** » **Ranges** toggles this filtering. But only some filters are applied by Grains (which ones???). Others can subsequently be applied manually.

Finally, **Segmentation Options » Clean Borders** = All discards regions bounded by image edges (segmented fibers clipped by image borders are incomplete and must be ignored).

The appearance of segmented regions can be altered in **Measurements » Options » Measurements Options » Appearance Options » Segmentation Options » Appearance**.

Grains App Files

- **Grains.IPX** – is a ZIP container from which project files are extracted when it is loaded. Grains is a generic open module, which can be viewed and modified using the Scripting Workbench.
 - **GrainsApp.VB** defines the UI (User Interface).
 - **ModuleGrains.VB** is the Visual Basic code that does the separation.
- **MuscleFibers.IQO** – contains all settings in the **Measurement** and **Count/Size** ribbons, including RGB intensity filtering. IQO files can be loaded and saved in **Measurements » Options » Segmentation Options**.
- **MuscleFibers.GRO** – contains Grains app parameters for segmenting muscle fibers. GRO files are loaded and saved in **Grains** (figure above).
- **MuscleFibers.ISG** – contains **Smart Segmentation » Recipe Options** (only needed to create the optional image mask).

Refining the Segmented Image

Contours found by Grains effectively separate adjacent fibers, but may have defects such as concavities, jags and “drips”. Unfortunately, Grains does not have contour smoothing options. We therefore scan through the image manually, joining fragments, splitting fiber groups & deleting non-fibers. Missing contours should be added only after contour smoothing (see below) because if they touch or overlap existing contours, smoothing will produce fiber groups, which would then need to be split, and distinguishing contour colors would be lost in the masking stage below.

Cleaning

- (1) Use **» Relative » Close** to join fragments. This should be done prior to smoothing or the smoothed contours will have concave cusps at the splits.
- (2) **» Split** contours as needed, prior to smoothing so that the edges revealed by splitting participate in the smoothing. Splitting may result in tiny, eg, single pixel, measurement objects, which are easily selected in the **Histogram**⁴ and deleted
- (3) Delete erroneous contours.
- (4) Save the histological image with the manually refined contours to a file with the “- **Seg1**” suffix.

Smoothing – mask & filter

Smooth the contours in two stages, doing about half the required smoothing in each. Replace each contour with the area contained, to yield an image upon which we can perform morphological operations:

- (1) **» Image » Mask** creates a binary mask separating the region inside each contour from that outside.

“Open” the image to reduce (but not eliminate) segmentation defects, being careful not to affect segmentation of the fibers themselves:

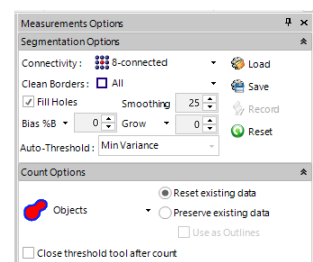
- (2) Refine the mask using **» Filters » 2D Filters » Morphological » Open: Shape = 2x2 Square, Passes ~ 20**. Verify that fiber areas are not altered by using the **Preview function: Apply filter to whole segment when fiber areas are optimized**.

Smoothing – built-in segmentation on mask

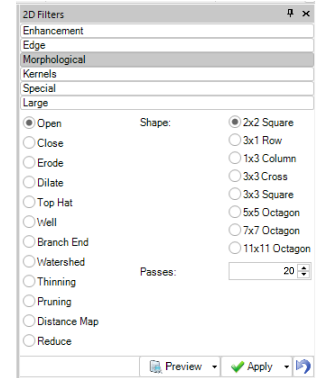
Use the built-in segmentation function on the mask image to find new contours:

- (1) Set **» Measurements Options » Segmentation Options » Smoothing ~ 25** (smoothing also tends to skip over “cracks” in fibers, although it creates unsightly concave cusps).

⁴ The **Histogram** can be used to filter measurement objects, which is not what we want to do here.



- (2) To compensate for reductions in area caused by smoothing (visible when measurement objects are pasted to the histological image, below), set » **Measurements Options » Segmentation Options » Grow** > 0. Use caution in setting grow value > 0 as increase in growth of the contours increases amount of single pixel measurement objects called holes.
- (3) Set » **Measurements Options » Segmentation Options » Fill Holes** = ✓ (smoothing also tends to skip over “cracks” in fibers, although it creates unsightly concave cusps).
- (4) Set » **Segment = Bright**. Click **Count** to create measurement objects corresponding to the white regions of the mask. Be sure to set » **Measurements » Ranges** = OFF or many objects will be filtered out.



Remove any holes in the measurement objects. Find them using the » **Views » Data Table**, making necessary repairs.

- (5) Delete the measurement objects found by Grains on the original histological image.
- (6) Set the active Class to **Algorithmic** and use the » **Features Manager** to collect the new measurement objects created on the mask (copy) and “set” (paste) them onto the original histological image.
- (7) Set the active Class to **Manual** to associate a label and color to manually added contours, providing some information about the manual labor required to segment the image (joining, splitting and deleting fail to be tagged as manual, making this an underestimate).
- (8) Add missing contours with » **Direct » Polygon**. Now, it is possible to create overlapping contours, which might be necessary if the tissue itself has a folding defect, without causing problems.

Any additional segmentation repairs can be done manually. If there are a large number of defects, it might be worthwhile to review the parameters of the preceding operations.

- (9) Save with “- **Seg2**” tag.
- (10) Review Segmentation Data by » **View » Histogram**. Observe all contours close to, and outside of limits to determine validity of contours and measurements.

Examine and Export Segmentation Data

Export the segmentation data table:

- (1) » **View » Histogram**
- (2) » **View » Data Table » Export » Text File**. Name this file with the suffix “- **Meas**”, as illustrated above.

Also export a version of the image with the measurement objects “burned in”, that is, made a part of the image, which is useful as an illustration. Name this file with the suffix “- **Burn**”, as illustrated above:

- (1) » **Measurements » Options » Appearance Options » Outlines » Width** = 8
- (2) » **Export » Snap**

Modifying the Grains App

Add-in modules are managed as “projects” in the development kit: Click **Macros » Scripting Workbench**.

- ⇒ **GransApp.vb** is the Grain app UI.
- ⇒ **ModuleGrains.vb** is the code.

