

Introduction to High Content Image Analyses in Large Scale Screening Studies

- Bruce Aronow
- Human Progenitor Cells Consortium (NHLBI)
- Insights : DOD / DTRA HTS Informatics
- Single Cell Genomic Atlas Gudmap/Facebase/Lungmap
- Bioinformatics Core Director
- Cincinnati Children's Hospital Medical Center

key work in here from Andrew Plassard and Shikha Chaganti; Computer Science, University of Cincinnati; Scott Tabar, Eric Bardes, CCHMC; and Thomas Lamkin and Steve Kawamoto, Air Force Research Laboratory, Wright Patterson AFB

Image Analysis

- **Insight** and understanding requires our ability to identify and integratively analyze what we see and perceive
- Even simple human tasks involves sophisticated **visual cognition** – eg reading, face recognition, driving a car, or examining a slide under a microscope.
- For the design of any intelligent system, it is vital to have a robust image analysis capability embedded within it.
- Much of research data is lost due to incomplete recognition, quantitation, and post-hoc analysis.

Bioimage informatics: Why do we need automated image analysis in biology?

Automation:

- Advances in automated microscopy produce large volumes of image data.
- Manual inspection of this data is slow and inadequate.

Completeness:

- Heterogeneous objects and object related features must be tabulated by objective/unbiased methods

Invisible patterns:

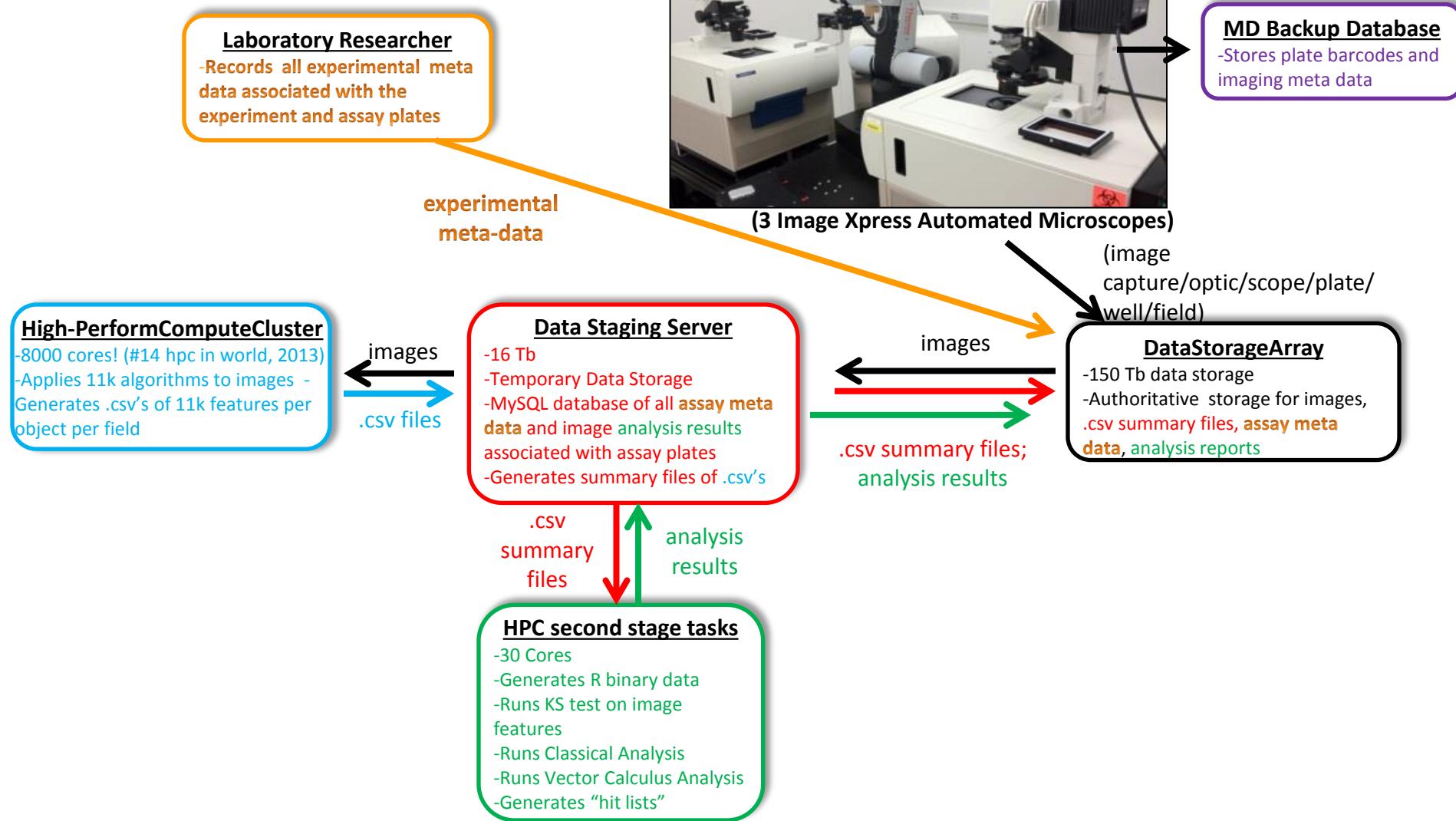
- High-Dimensional Data Mining Techniques are required to detect and analyze relationships between object classes whose differential and correlated occurrence defines the composition and state of a complex structure

Image Analysis

- **Insight** and understanding requires our ability to identify and integratively analyze what we see and perceive
- Even simple human tasks involves sophisticated **visual cognition** – eg reading, face recognition, driving a car, or examining a slide under a microscope.
- For the design of any intelligent system, it is vital to have a robust image analysis capability embedded within it.
- Much of research data is lost due to incomplete recognition, quantitation, and post-hoc analysis.

Large Scale Cell Effects-Profiling by High Content Image Capture and Analysis

Hardware, network and data capture pipeline

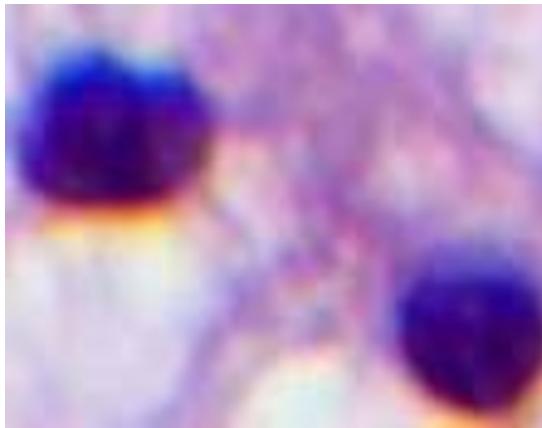


Segmentation Approaches

Iterative Thresholding

Cleaving

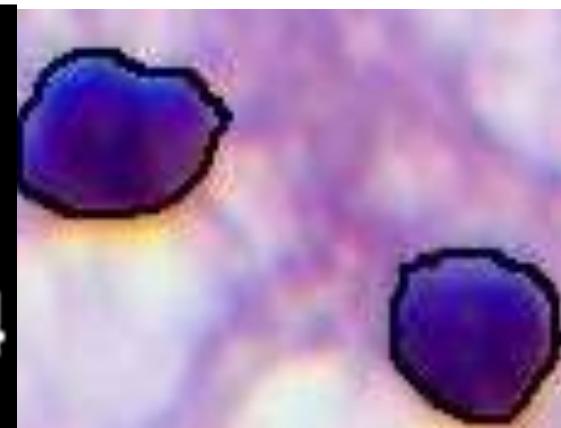
- **Segmentation** - The fundamental step in quantitative image analysis is the identification of objects and structures within a scene or image



Original image



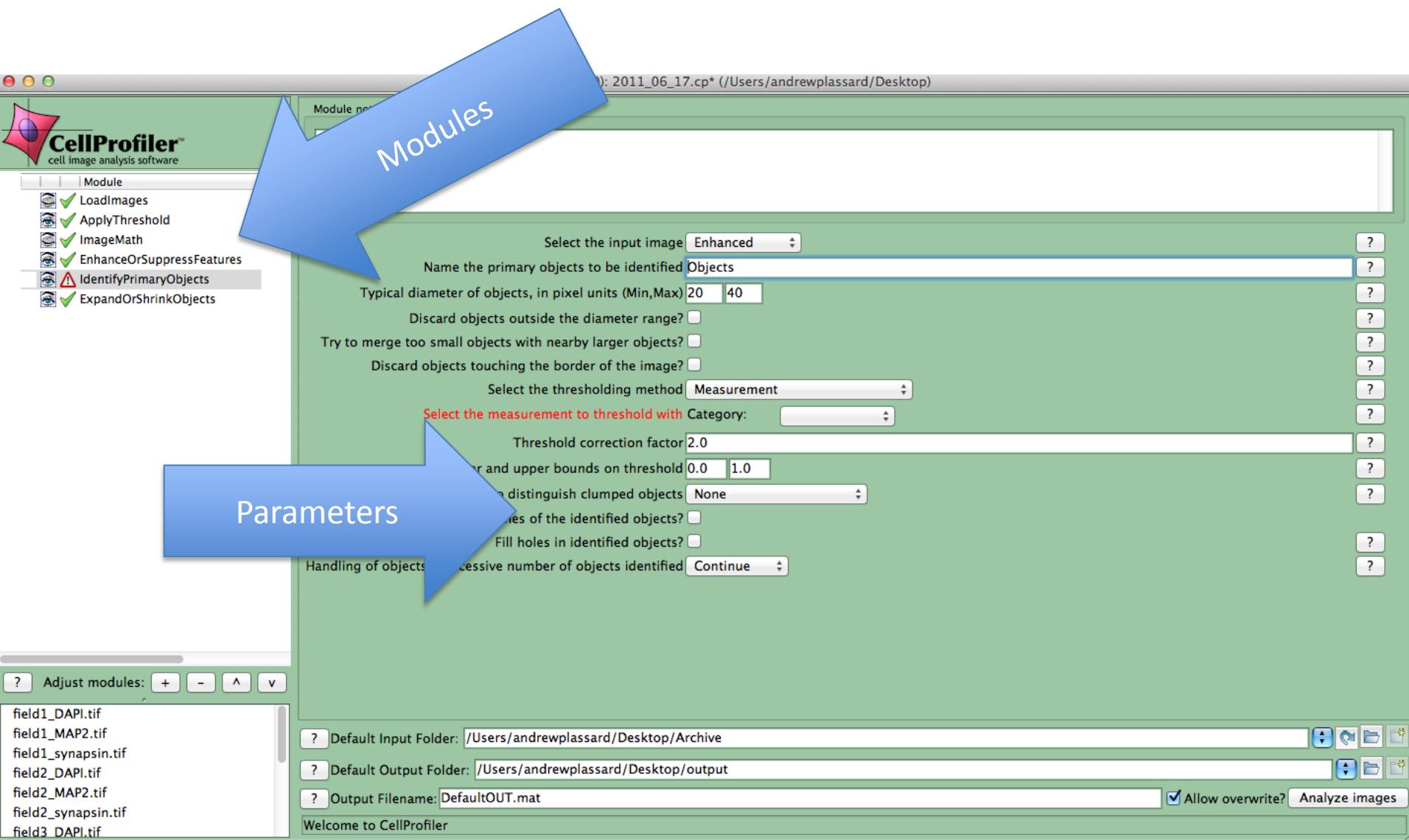
Mask



Object outlines

- The **shape, size, count, location, intensity, and texture** of these structures provide vital information about the presence and extent of a disease.
- It is very important to identify **accurate locations** of cells to study tissue histology or architecture.
- Popular methods for cell segmentation: watershed, active contours, thresholding

Using Cell Profiler™ as a preliminary means of masking and measuring principle features in microscopic cell-based imaging



Cell Profiler™ enables a biologist to adjust shape, color, shading parameters using differential thresholding settings. Doing this in “training images/examples” allows these principle components of microscopic images to be consistently identified and then quantitated across many wells, plates, batches, and studies

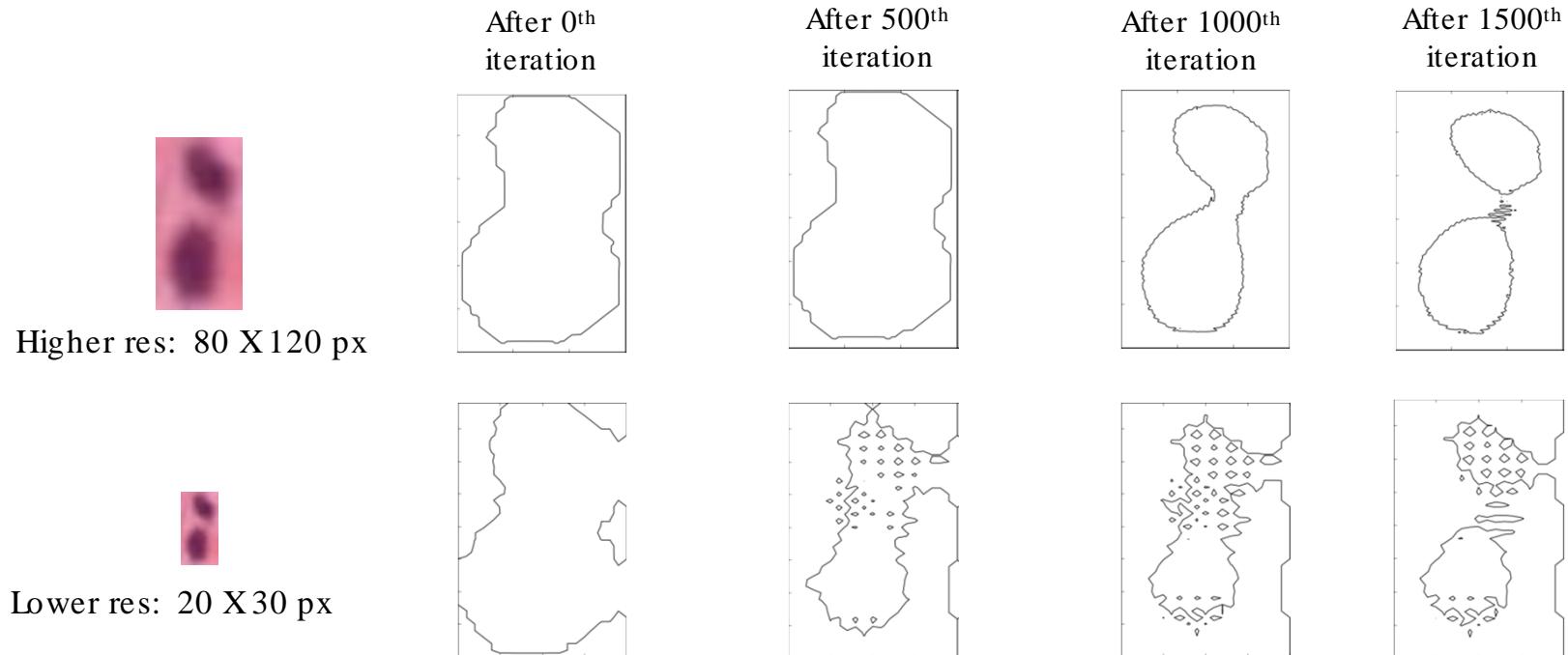
Example Threshold Options

Different Threshold Types

- Otsu Global
- Otsu Adaptive
- Otsu PerObject
- MoG Global
- MoG Adaptive
- MoG PerObject**
- Background Global
- Background Adaptive
- Background PerObject
- RobustBackground Global
- RobustBackground Adaptive
- RobustBackground PerObject
- RidlerCalvard Global
- RidlerCalvard Adaptive
- RidlerCalvard PerObject
- Kapur Global
- Kapur Adaptive
- Kapur PerObject
- Manual
- Binary image
- ✓ Measurement

Segmentation Approaches**Iterative Thresholding****Cleaving**

Example of level set method - an active contour method with evolving interfaces/boundaries.



Works better on higher resolution images than lower resolution images.
This is true of all watershed and active contours methods.

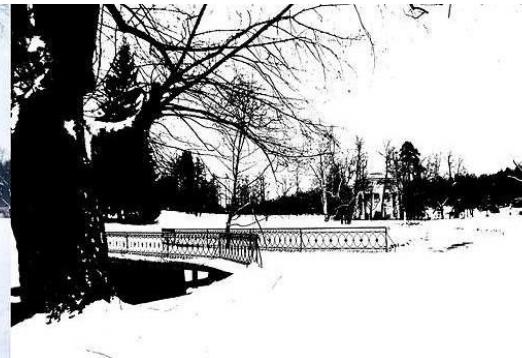
Segmentation Approaches**Iterative Thresholding****Cleaving**

Thresholding: converts an image into a binary image containing a **foreground** and **background** using one or multiple thresholds.

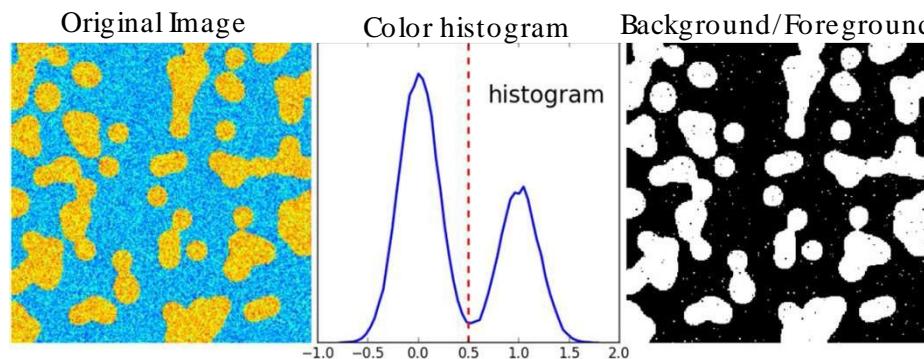
Color image



Thresholded image



© AKA MBG, composite by w: user:Nauticashades [Public domain], via Wikimedia Commons



http://scipy-lectures.github.io/advanced/image_processing.html. Histogram based thresholding.

Segmentation Approaches

Iterative Thresholding Cleaving

Otsu Thresholding: The Otsu method computes a global threshold T , that minimizes the within-class variance a^2 , given by,

$$a^2(T) = w_b a_b^2 + w_f a_f^2$$

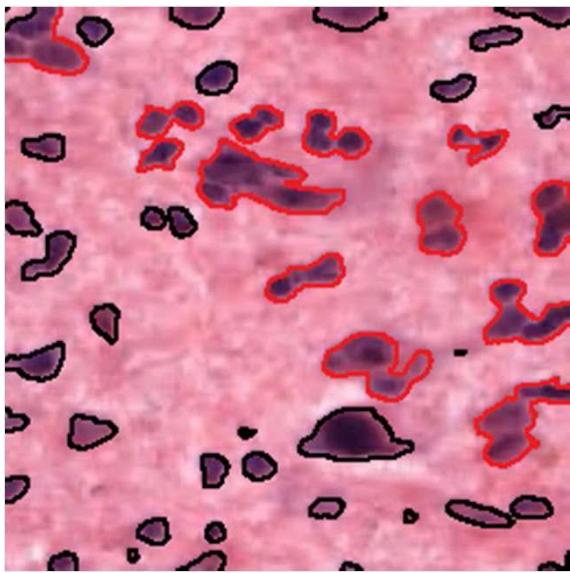
where a_b and a_f are standard deviations of the background and foreground classes respectively and,

$$\sum_{i=0}^{L-1} w_i p(v_i); w = \sum_{i=T}^{L-1} p(v_i)$$

Where $p(v)$ is the probability that a given pixel in the image has a color v .

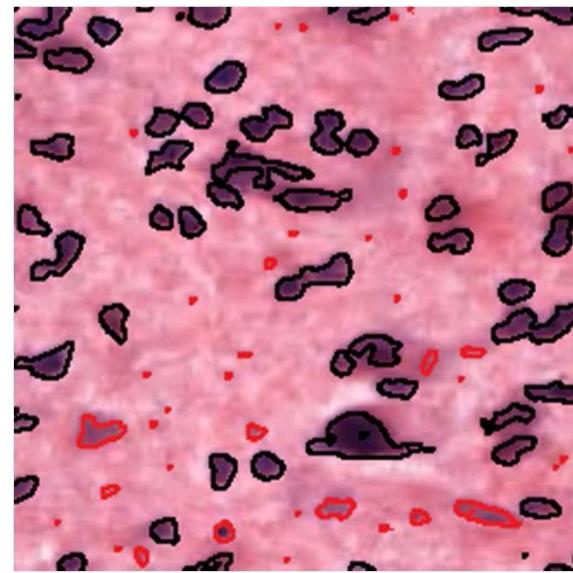
Adaptive Otsu: The adaptive Otsu method splits the image into overlapping windows and calculates threshold at each point

Boundaries identified by Otsu



- Uses a **global threshold**
- Insensitive to local changes in color (**red outlines**)
- Does not identify objects with weak boundaries

Boundaries identified by Adaptive Otsu



- Uses a **local threshold** for each window
- Not *selective* in choosing local region
- Misidentifies a lot of background artifact as objects (**red outlines**)

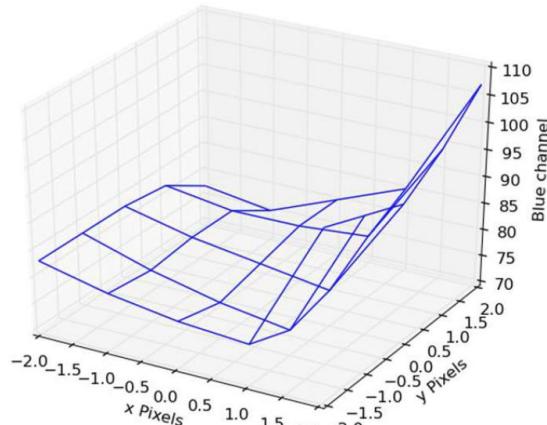
Trying to Overcome the limitations of adaptive Otsu: Iterative Thresholding

Image Preprocessing Step 1:

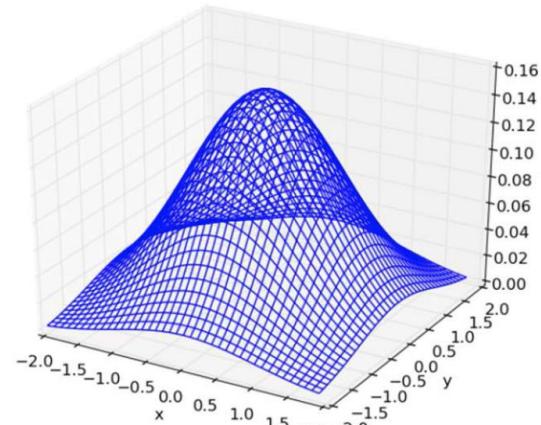
Gaussian convolution (C): It convolves the image I , with Gaussian kernel G ,

$$C(x, y) = I(x, y) * G(x, y) = \sum_{s=-\frac{k}{2}}^{\frac{k}{2}} \sum_{t=-\frac{k}{2}}^{\frac{k}{2}} f(s, t)g(x - s, y - t)$$

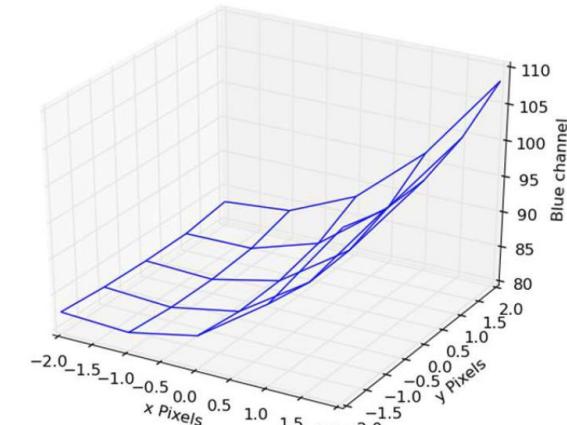
It has the effect of smoothing the image, by changing a pixel value to the weighted average of its neighbors.



Region with $K=4$



Gaussian kernel with $\mu = 0$ and
 $\sigma = 1$



Smoothed region

Segmentation Approaches

Iterative Thresholding

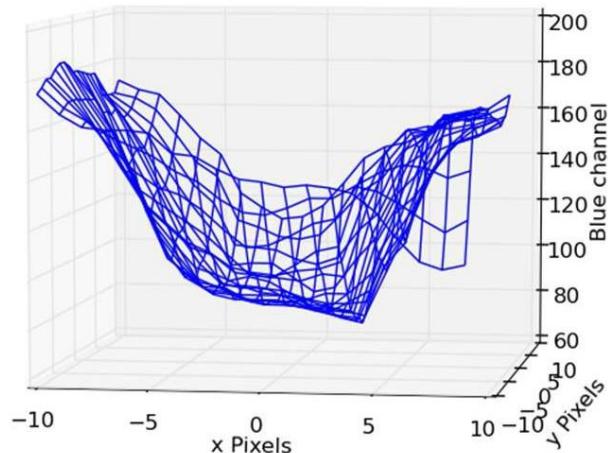
Cleaving

Image Preprocessing Step 2:

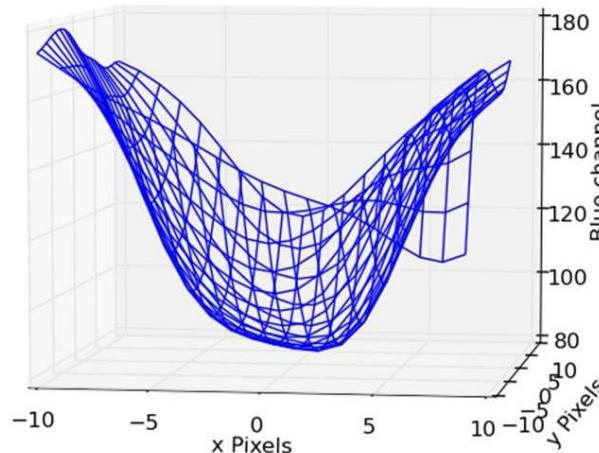
Gaussian normalizing filter (GNF): It is applied to the smoothed image, $C(x, y)$ obtained in the previous step to rescale it such that each pixel belongs to $[0, 1]$

$$F(x, y) = \frac{1}{1 + C(x, y)}$$

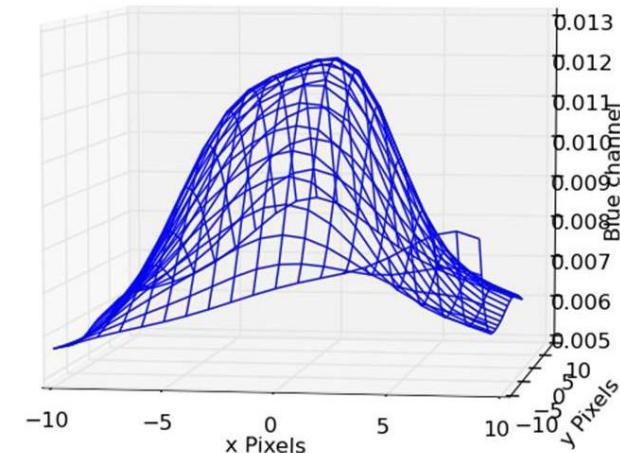
It changes the shape of the contour such that there is a peak over each object.



Original color distribution over an object



After Gaussian smoothing



After GNF

Algorithm: Iterative Thresholding

$F = getFilter(I)$

$mask = segment(F)$

Function **$getFilter(I)$**

$$return \frac{1}{1 + G * I}$$

Function **$segment(r)$**

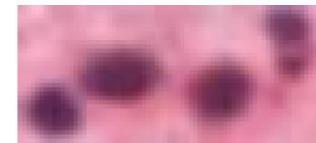
$r_{new} = r > (\mu(r) + \sigma(r))$

If $countSubregion(r_{new}) > 1$ then

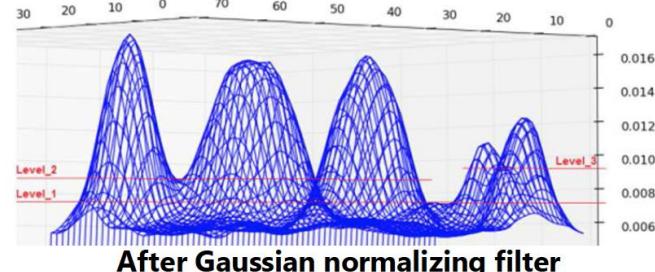
For s in $getSubregion(r_{new})$ do
 $segment(s)$

Else

STOP



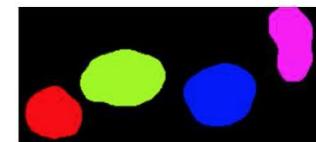
Original Image



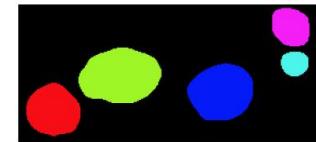
After Gaussian normalizing filter



$$t_1 = Lev^e_1$$

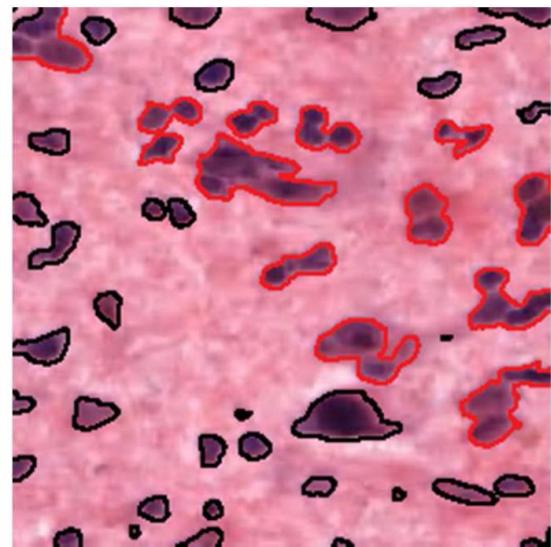


$$t_2 = Lev^e_2$$

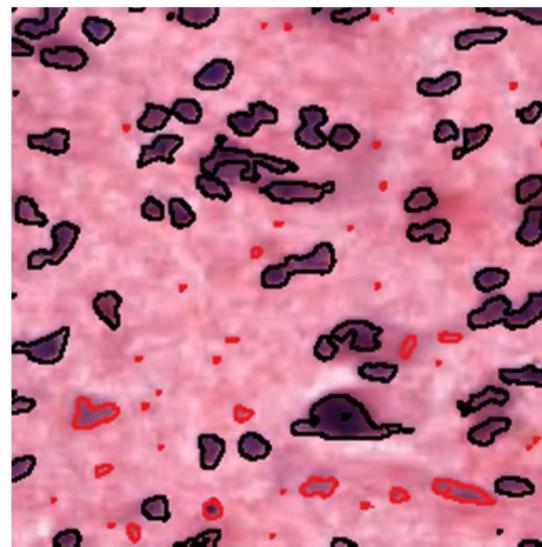


$$t_3 = Lev^e_3$$

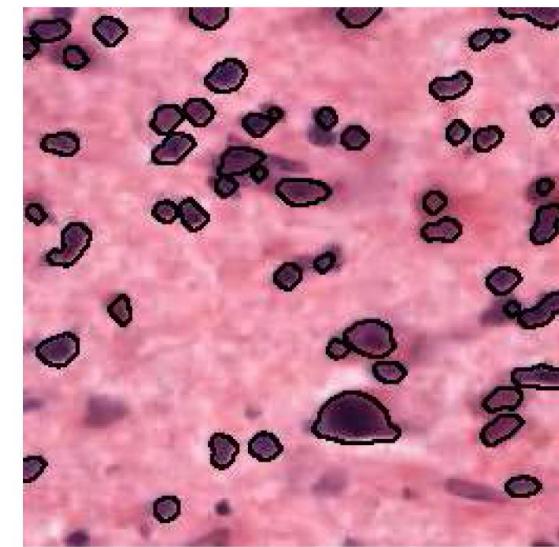
Comparison between the three thresholding approaches



Otsu thresholding

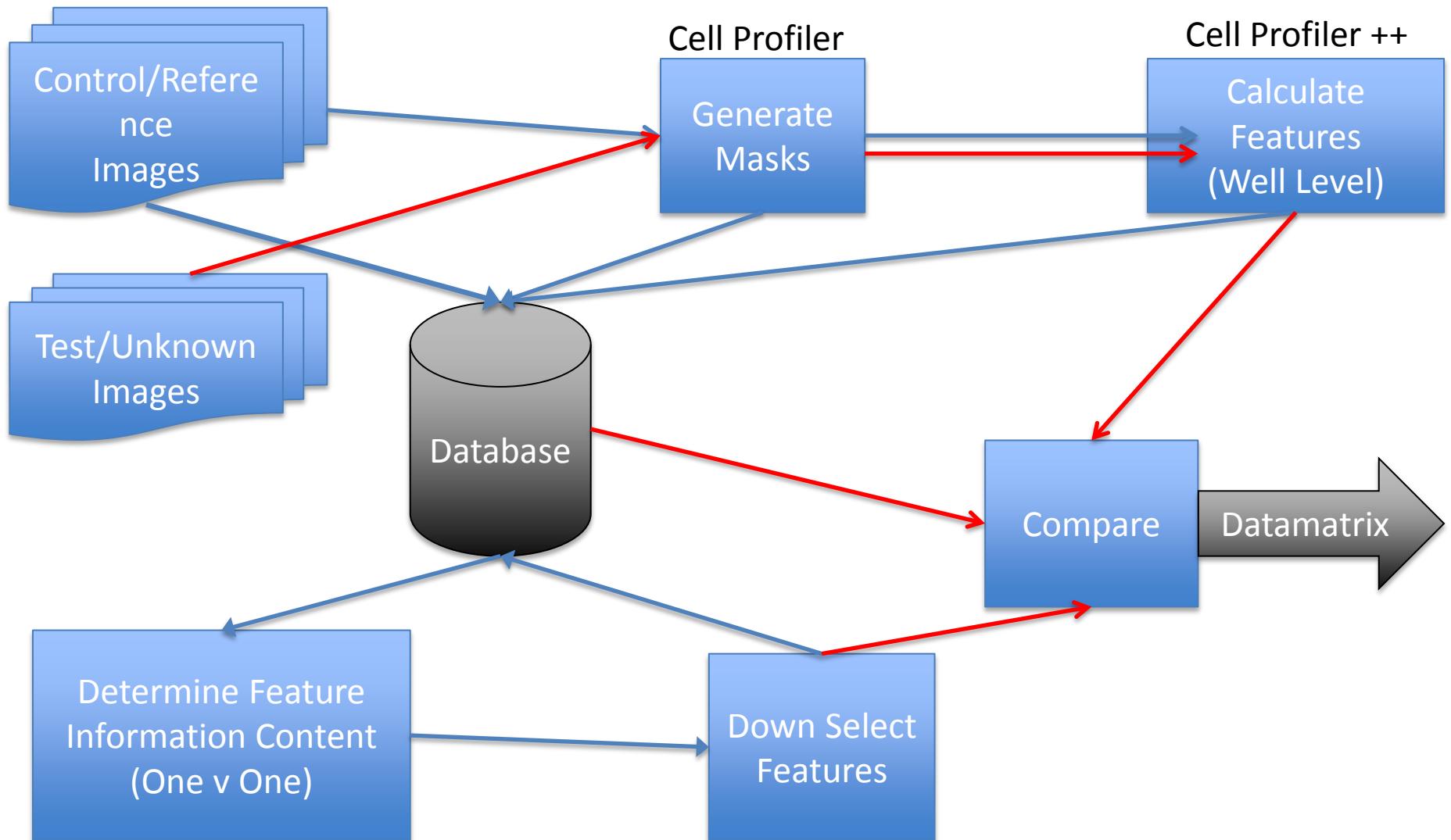


Adaptive Otsu

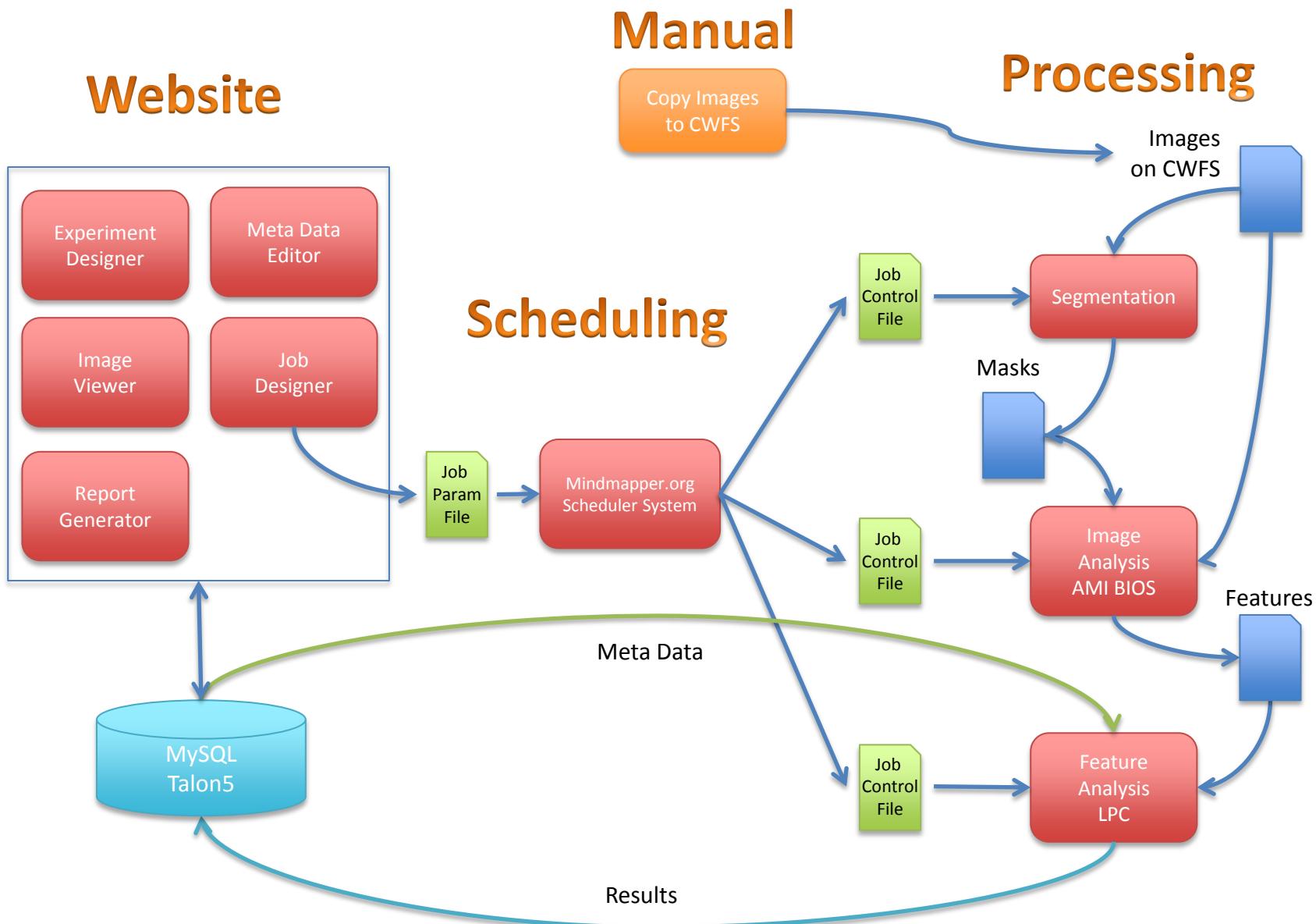


Iterative thresholding
Current method

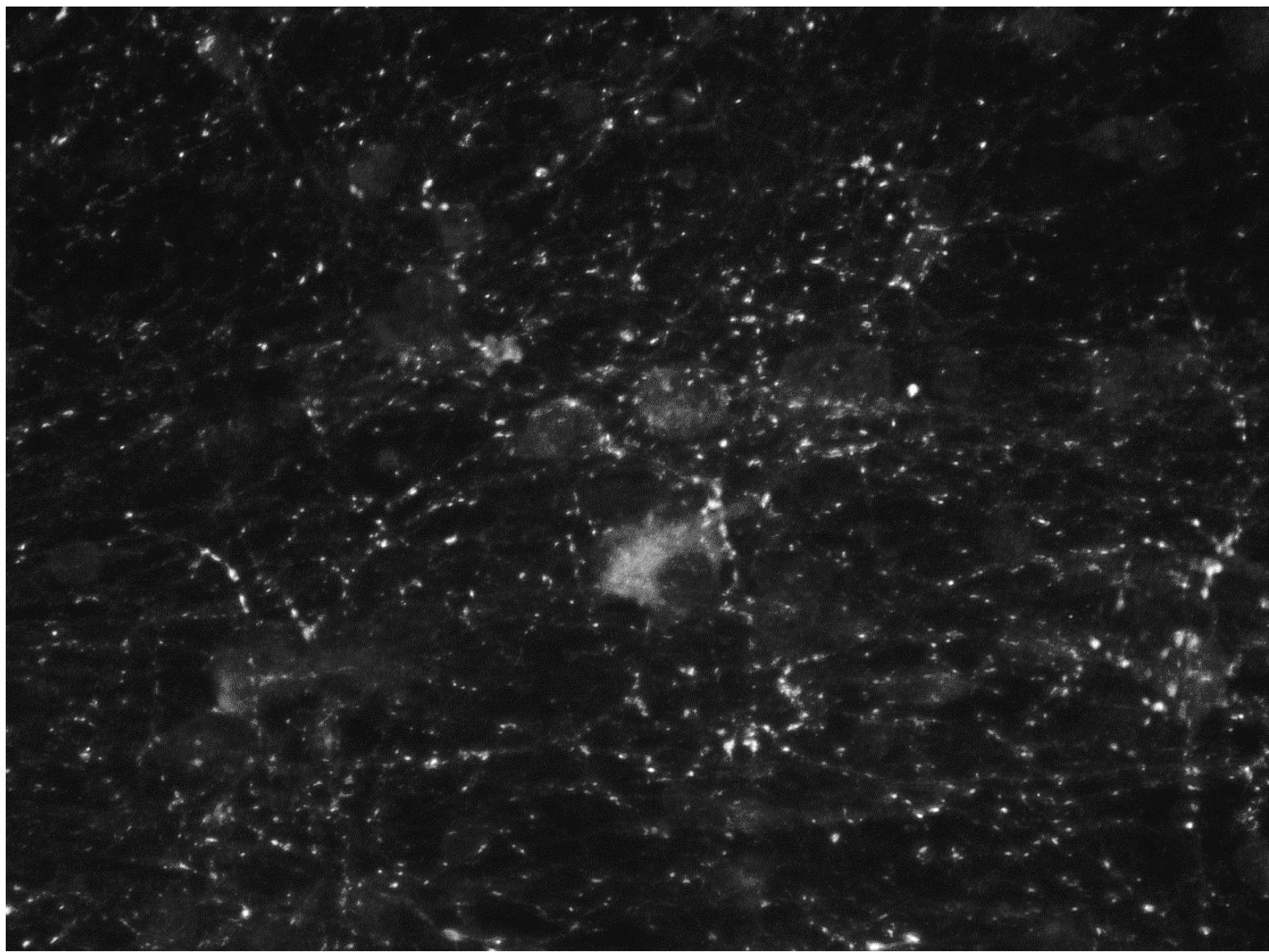
Iterative thresholding overcomes the problems of Otsu and adaptive Otsu.



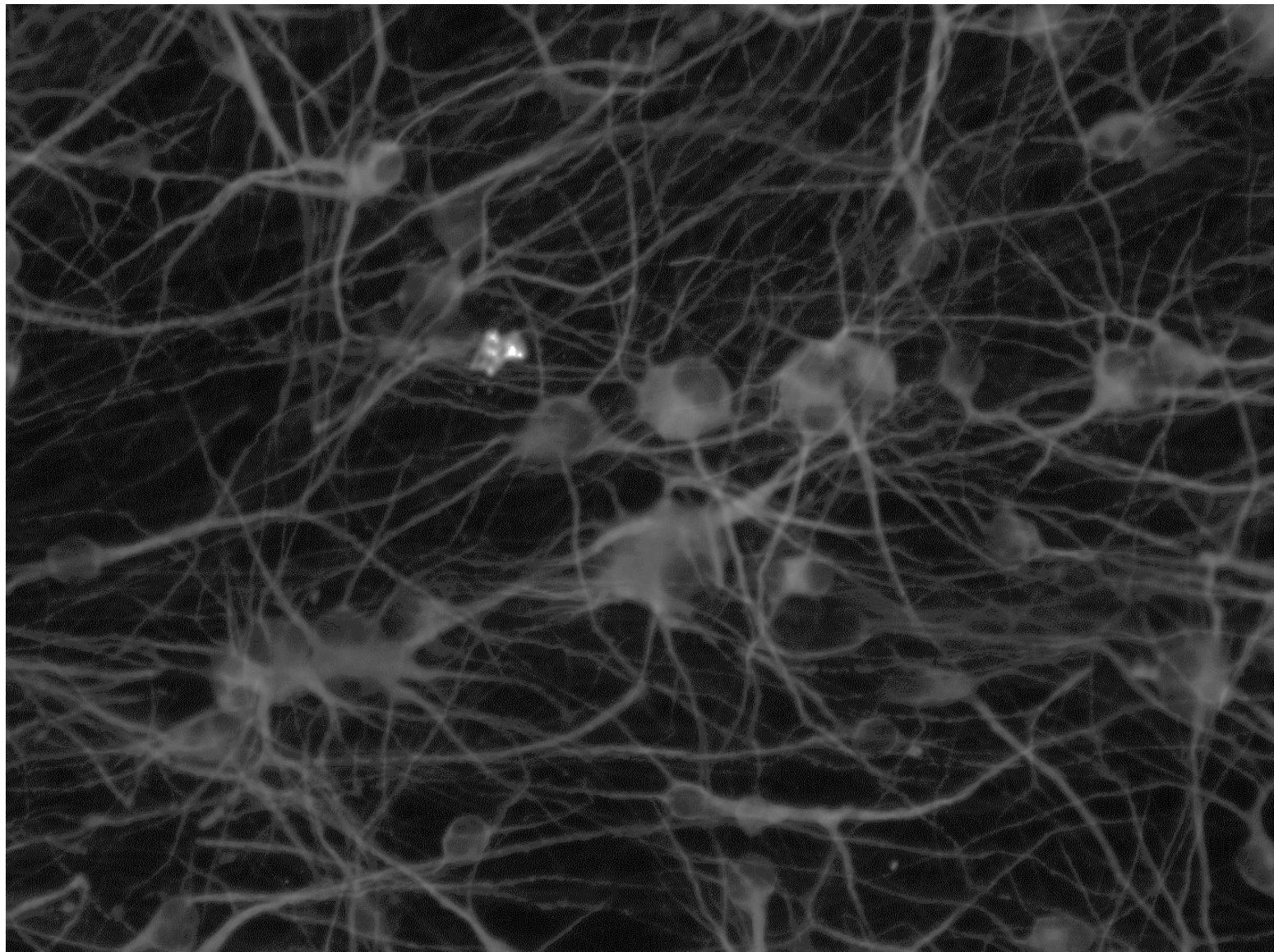
High Level Overview



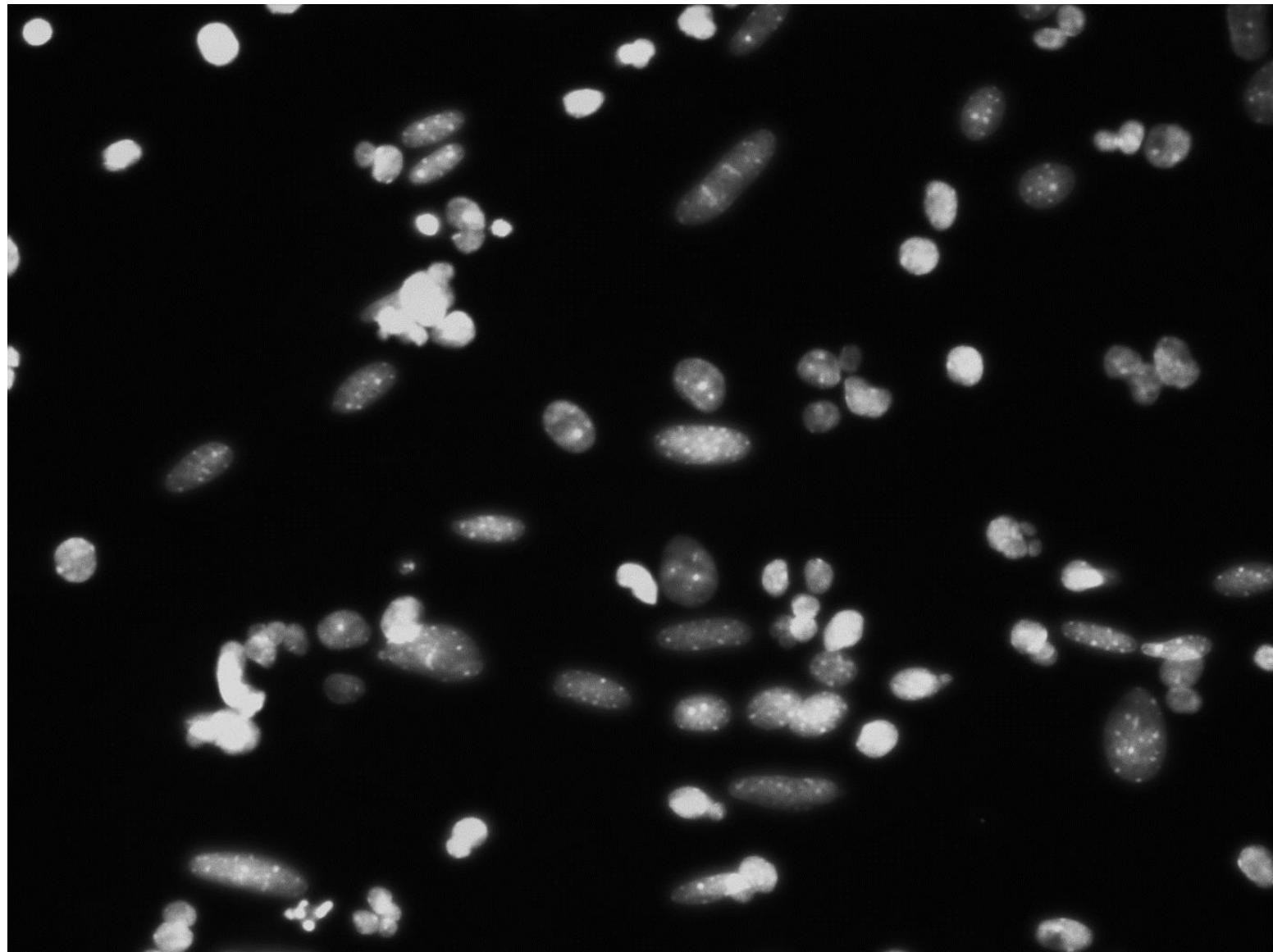
Assembling multi-channel images— synapses/synapsin R



Assembling multi-channel images— axons/MAP2

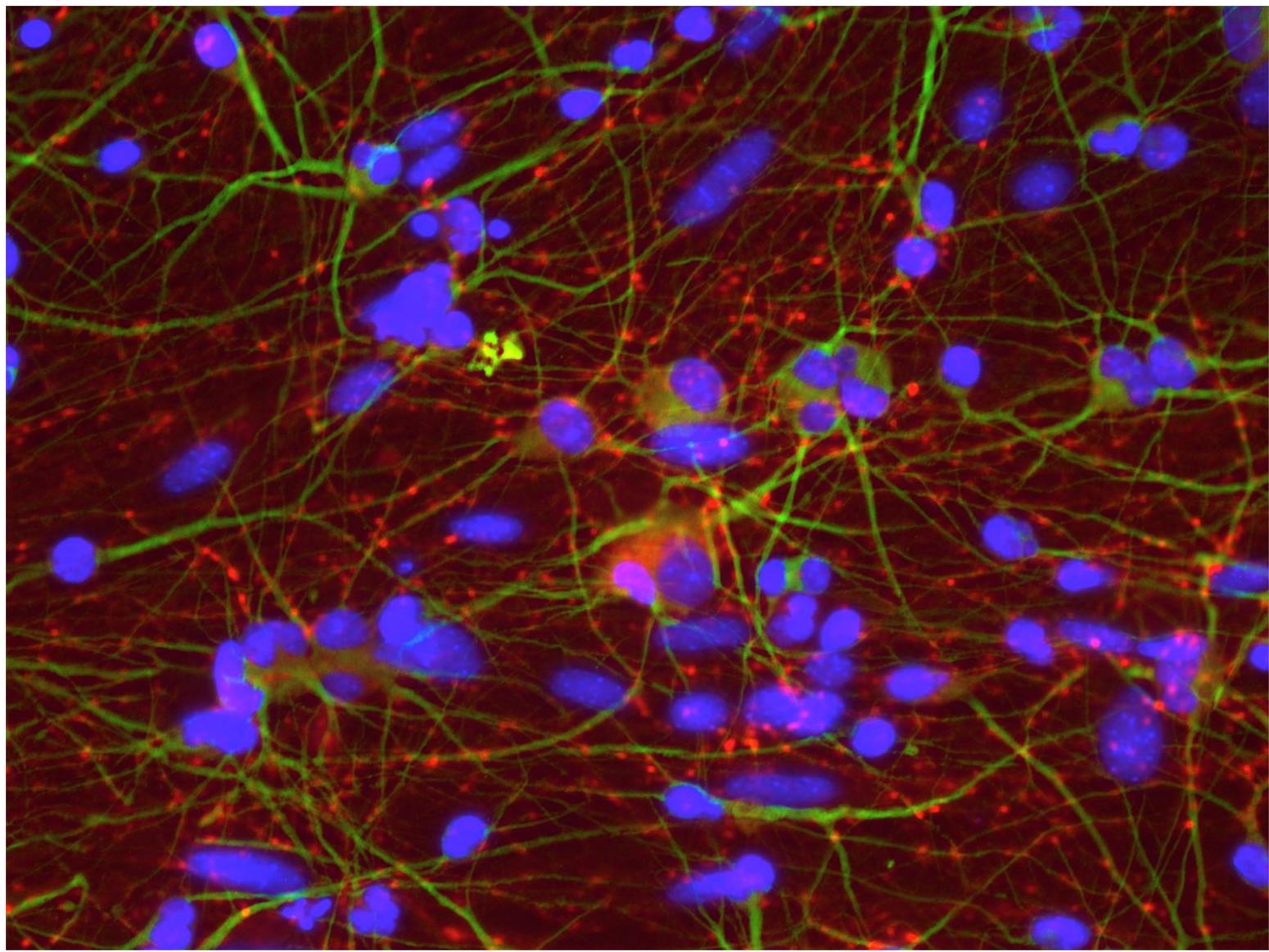


Assembling multi-channel images— nuclei/DAPI

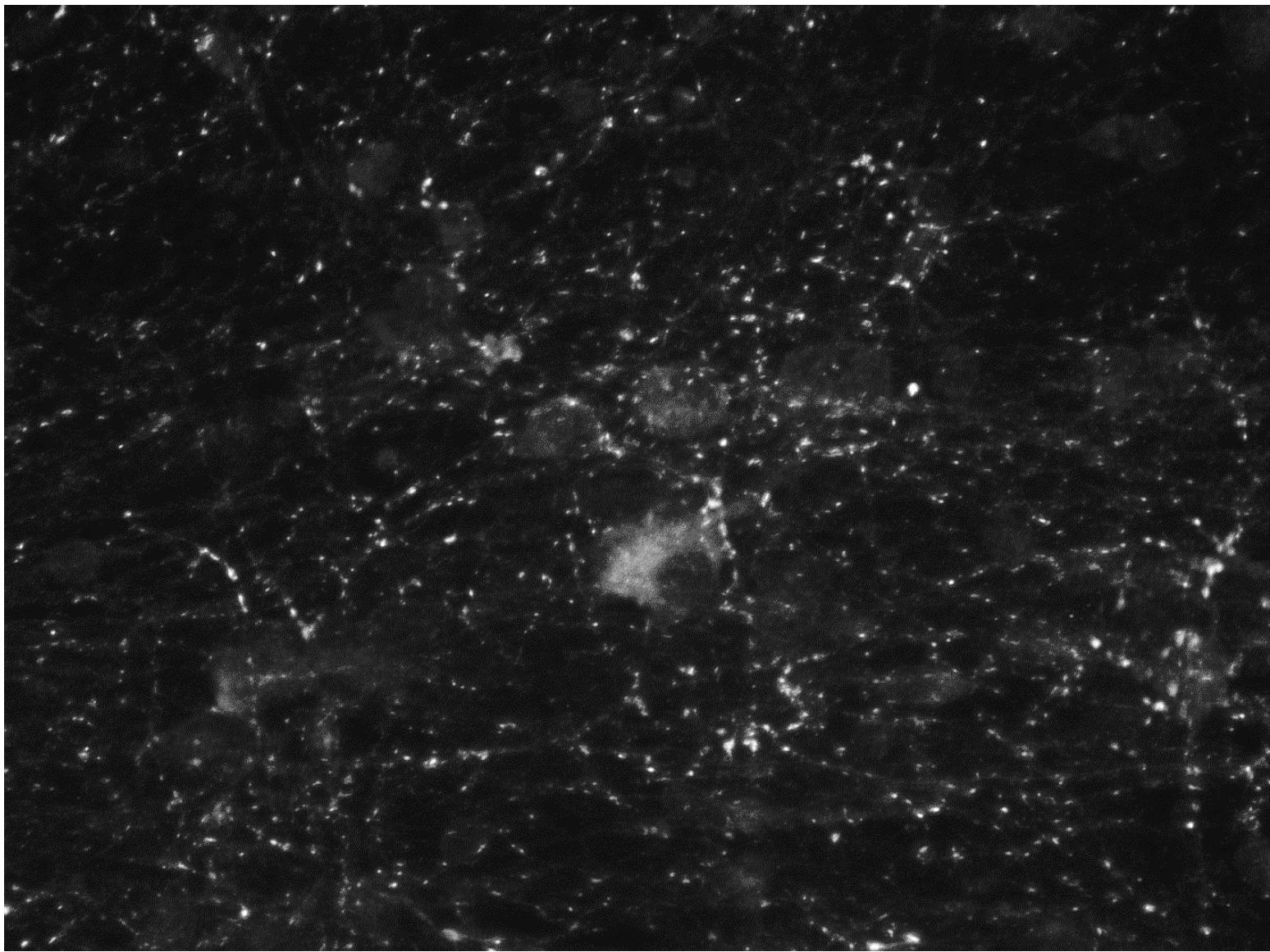


Composited multichannel image

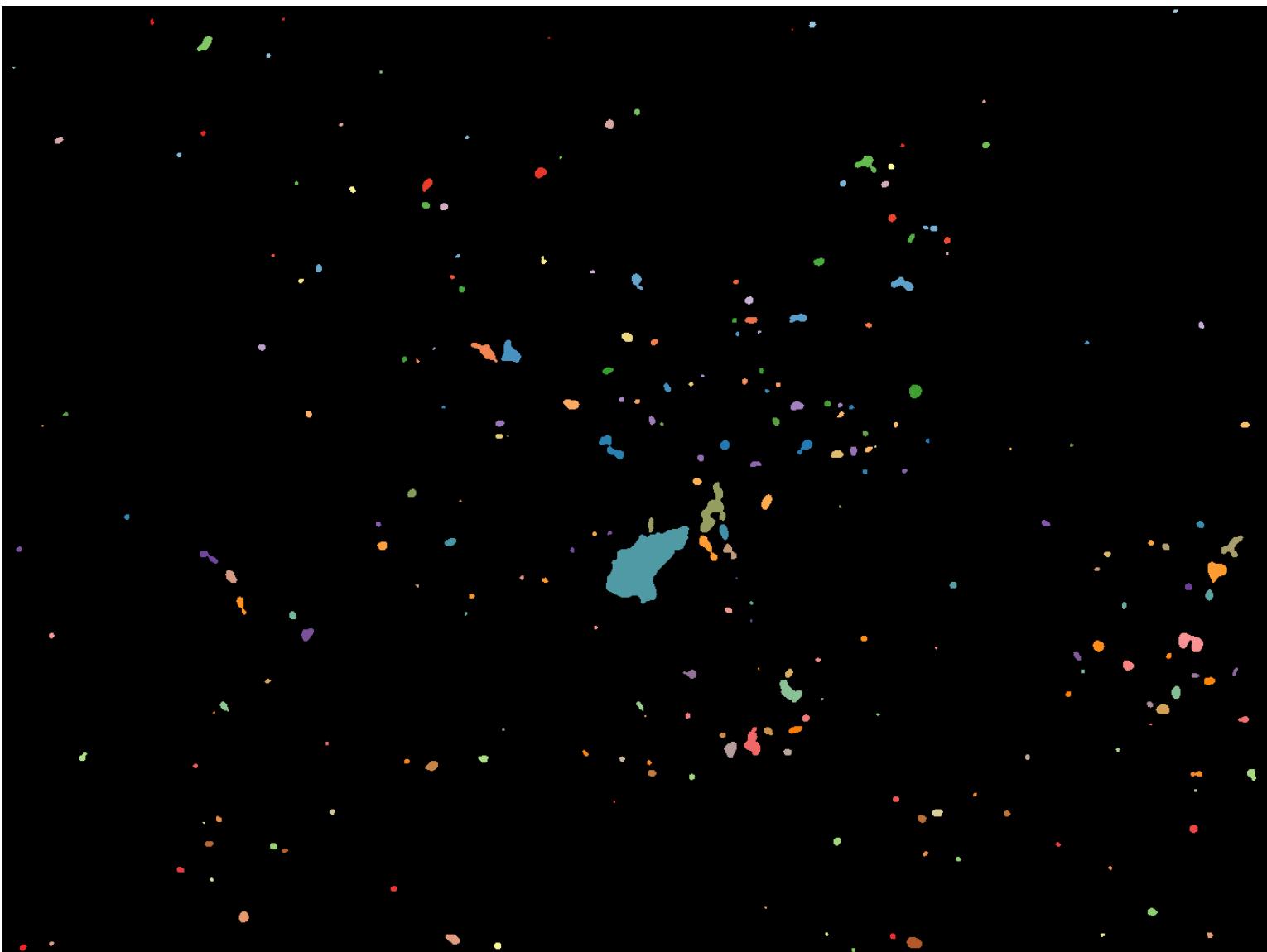
nuclei blue/axons green/synapses red



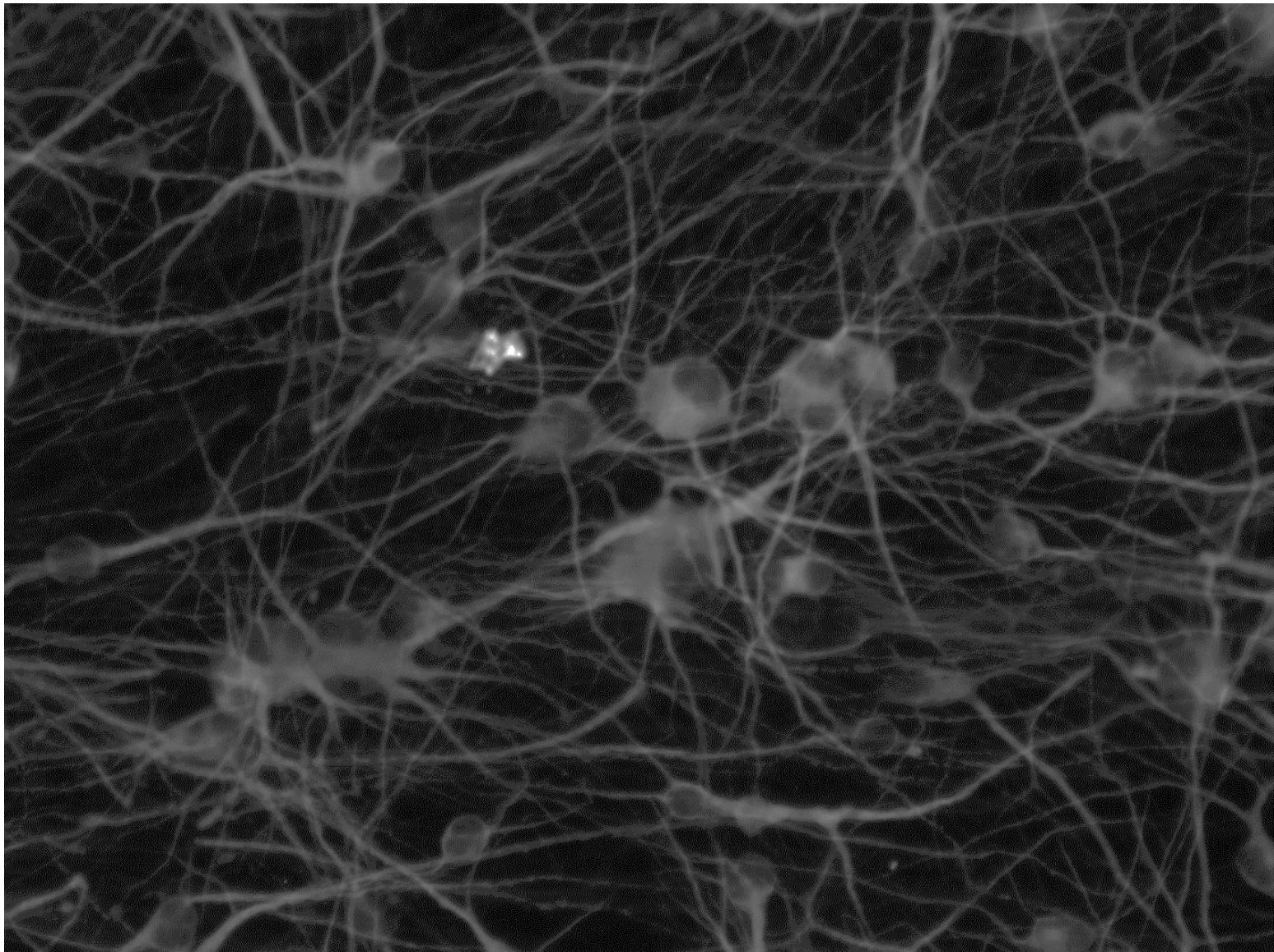
Segmentation– using multichannel images—synapsin/Red



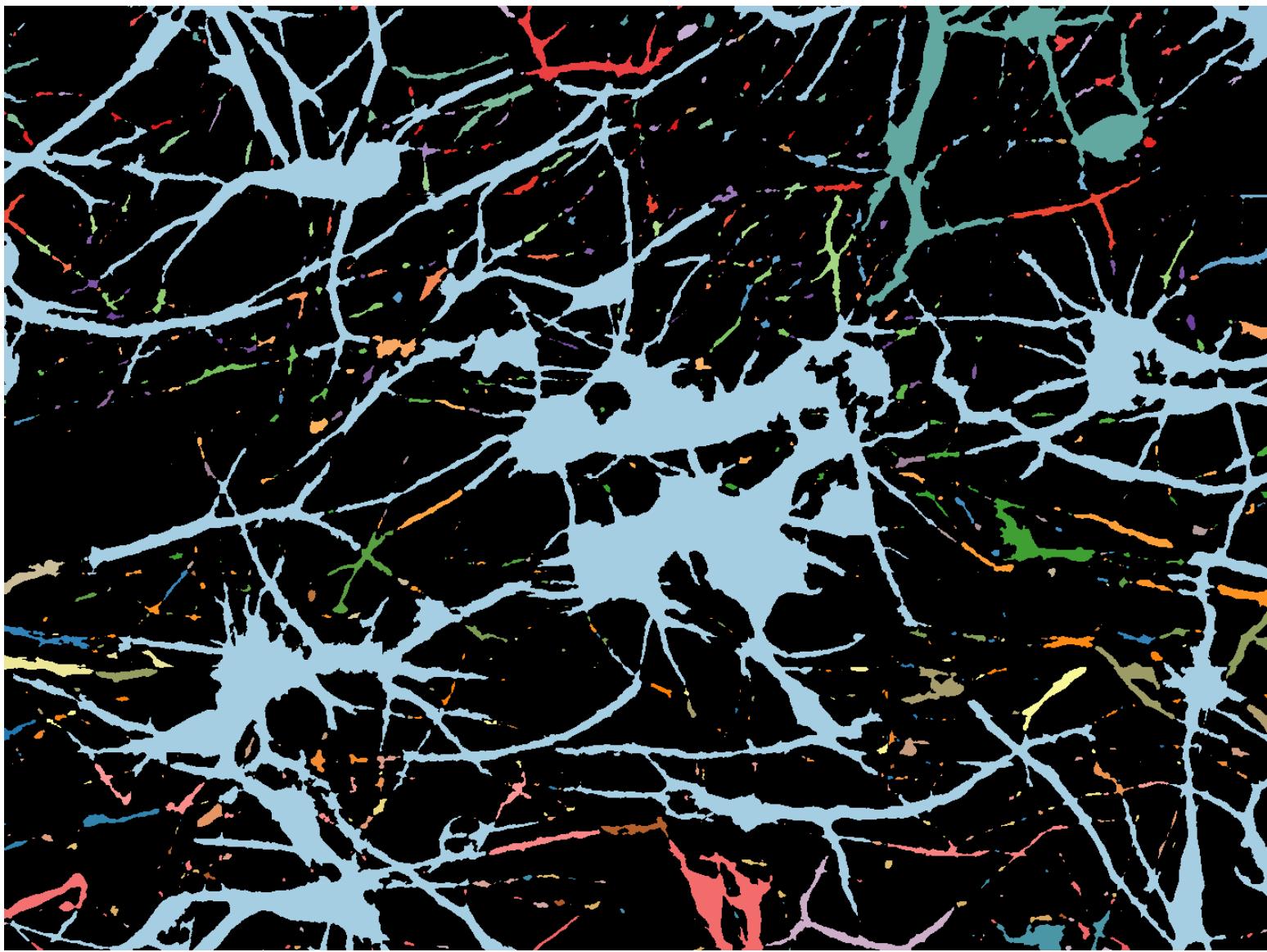
Segmentation– using multichannel images—synapsin/Red



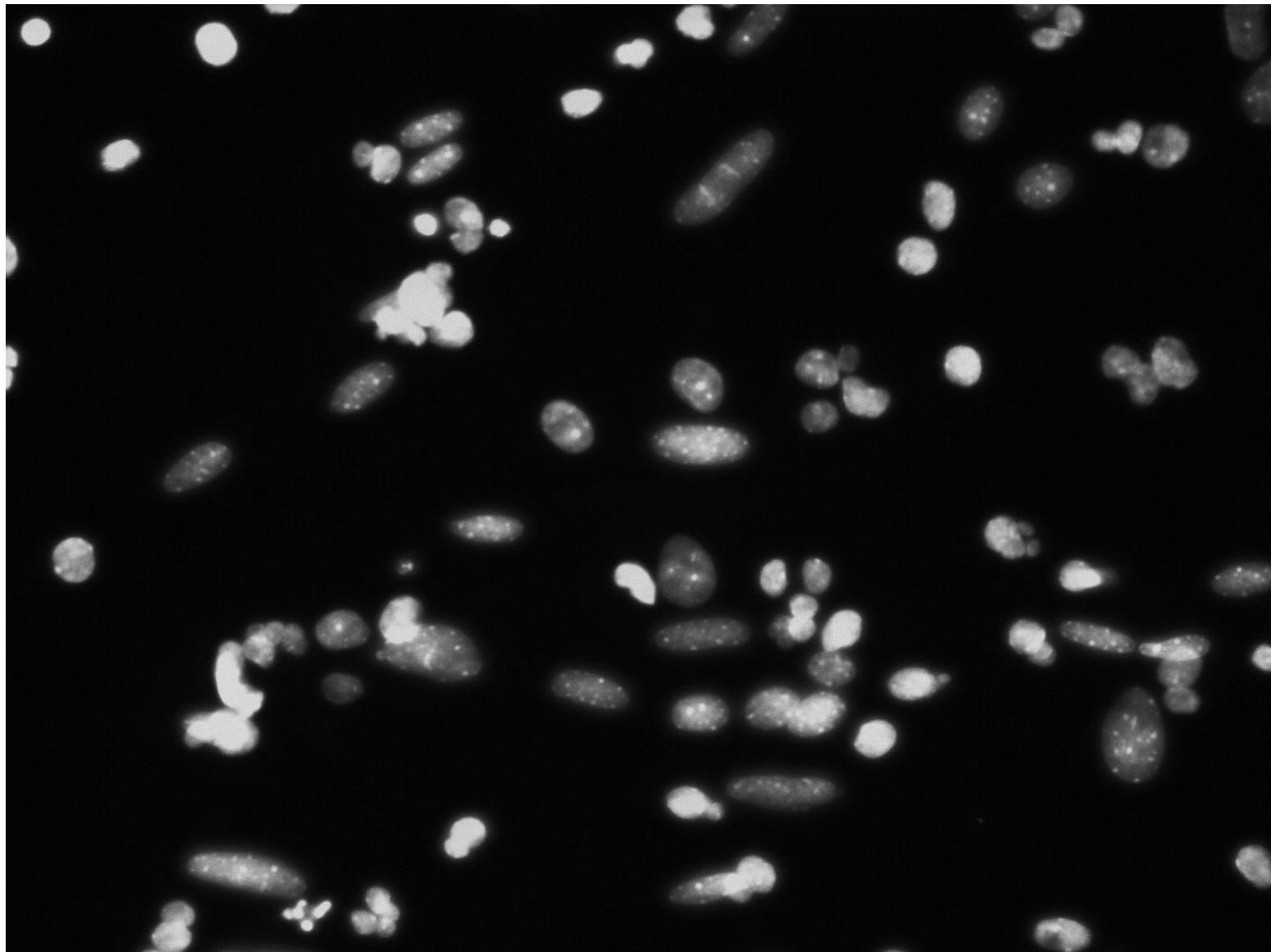
Segmentation– using multichannel images—axons/Green



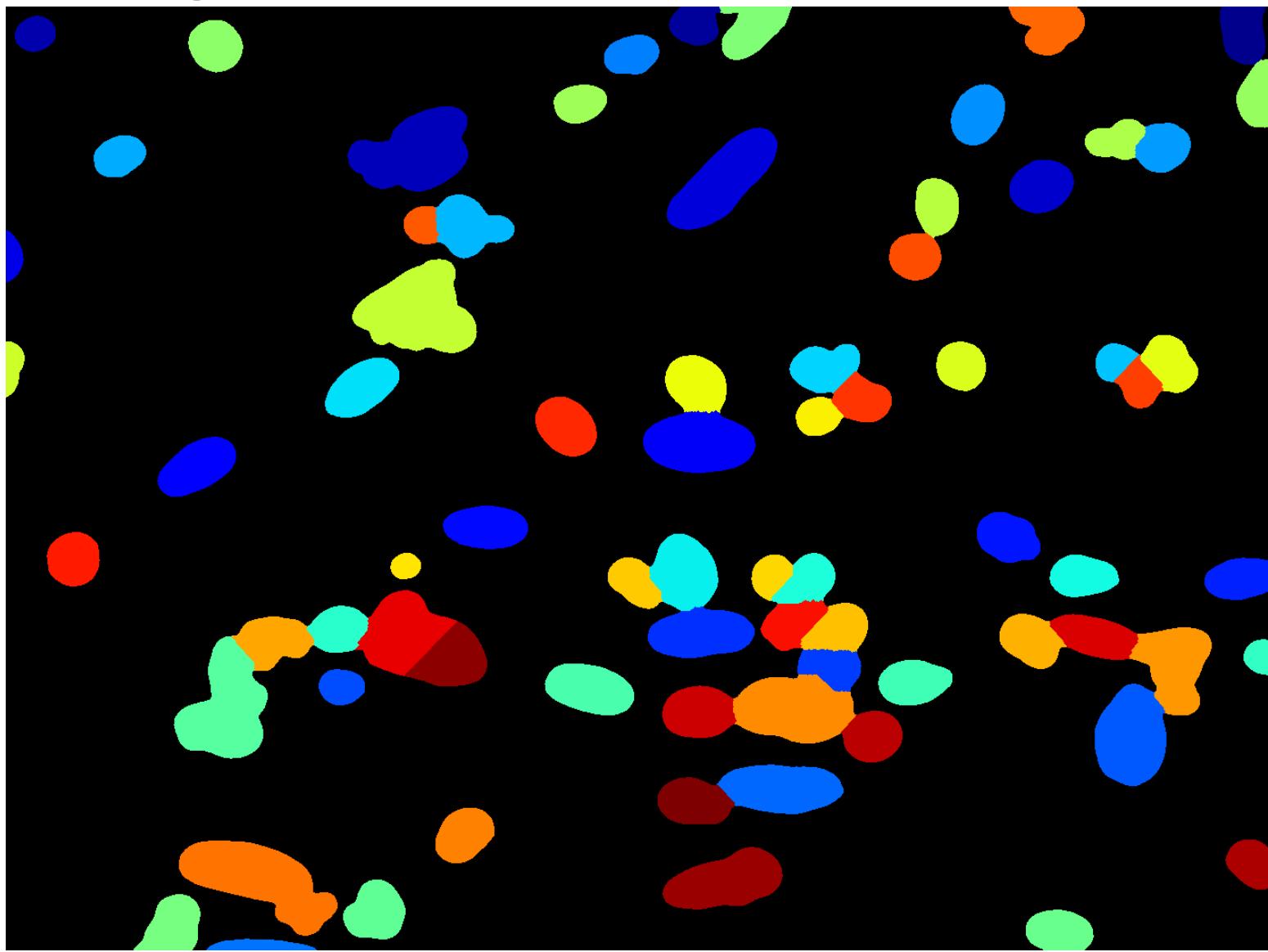
Segmentation— using multichannel images—axons/Green



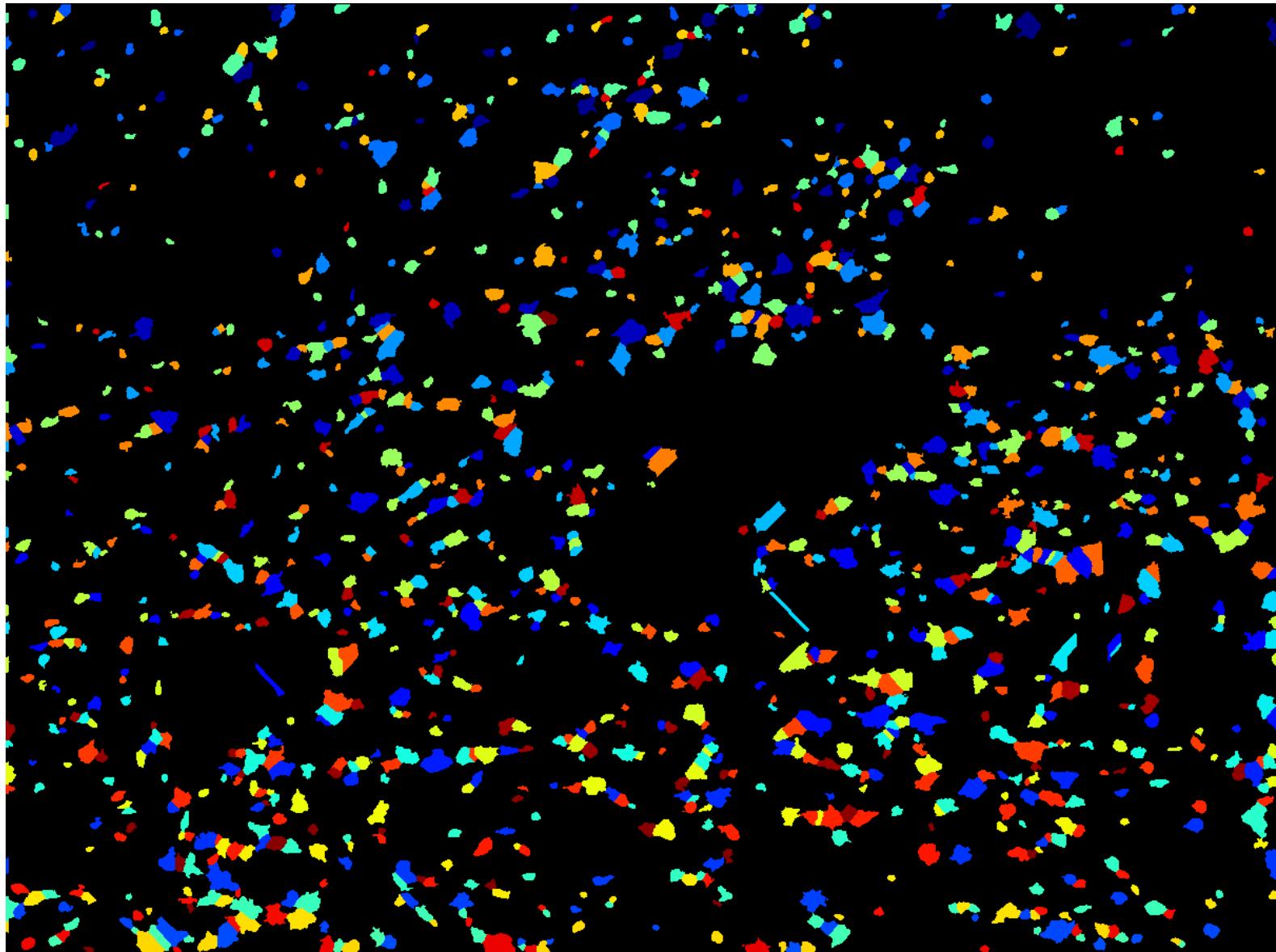
Segmentation– using multichannel images—nuclei/DAPI



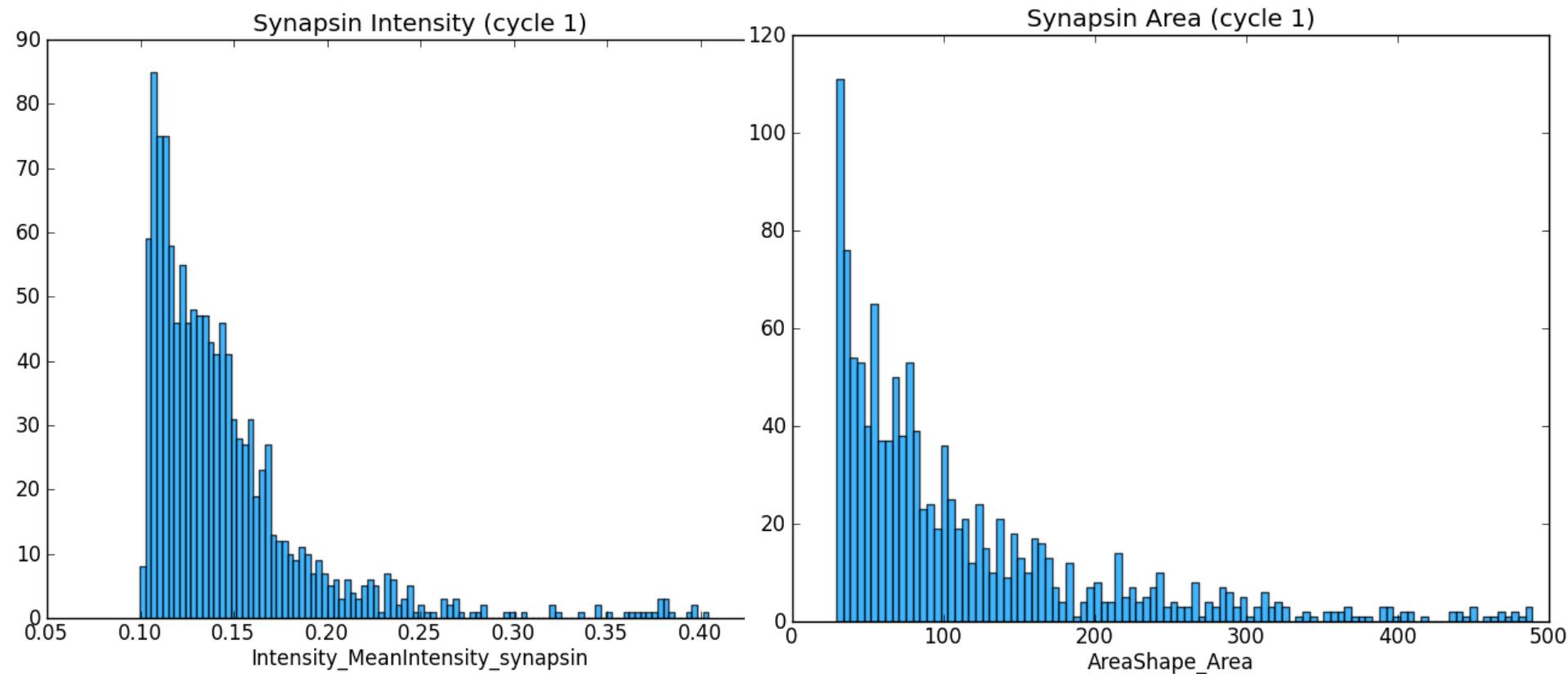
Segmentation— using multichannel images—nuclei/DAPI



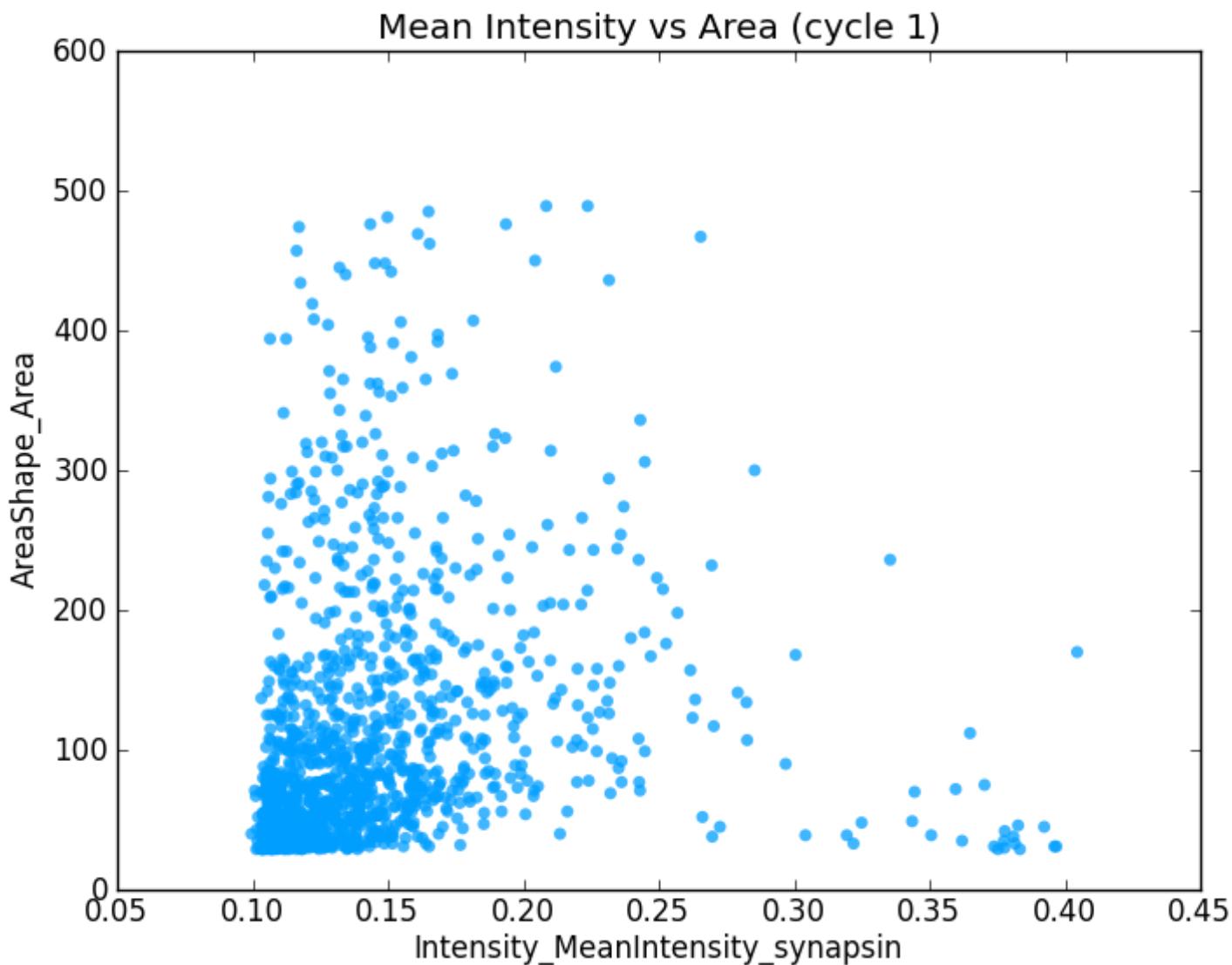
Synapsin objects



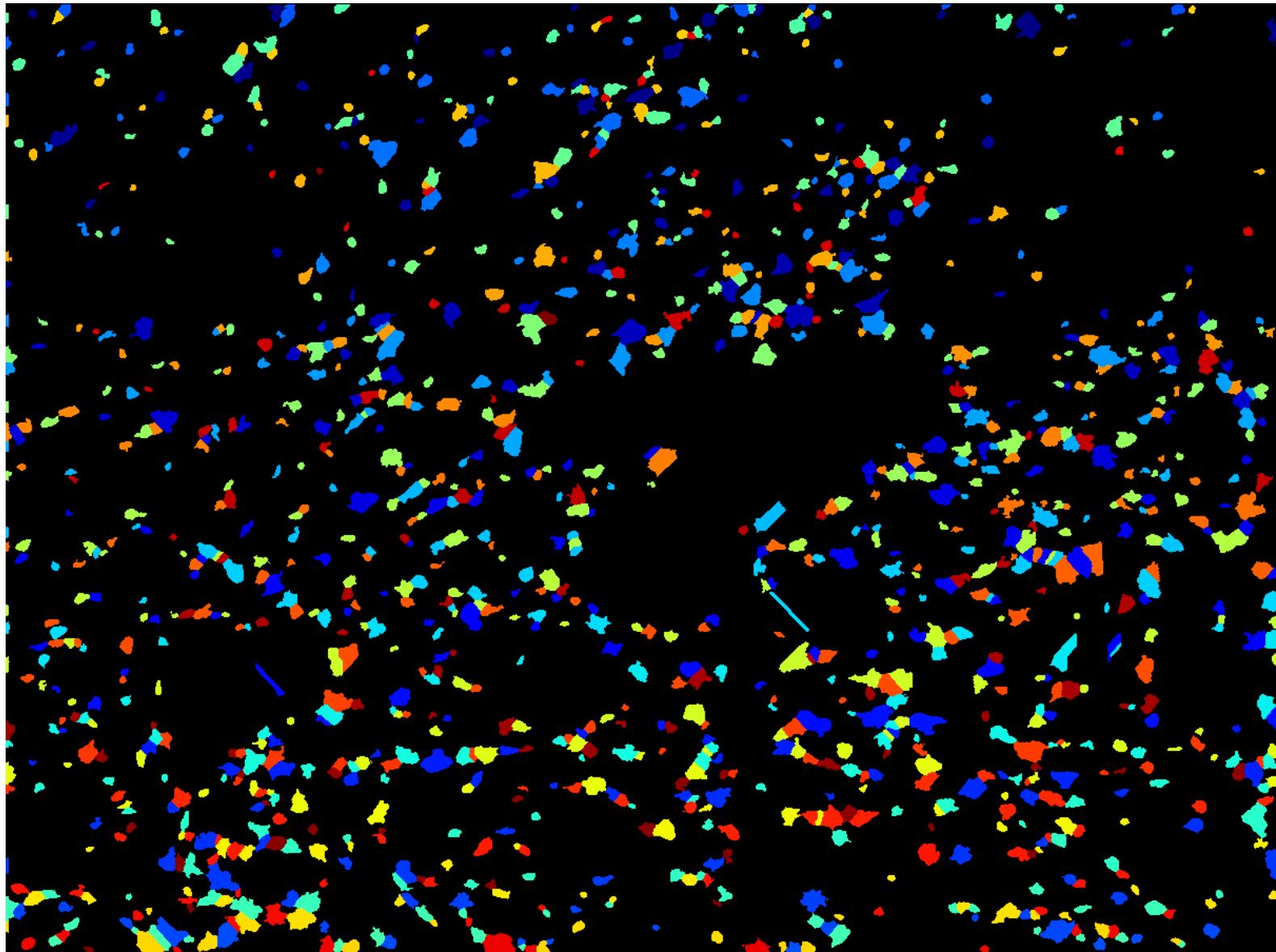
Synapsin Intensity Distribution



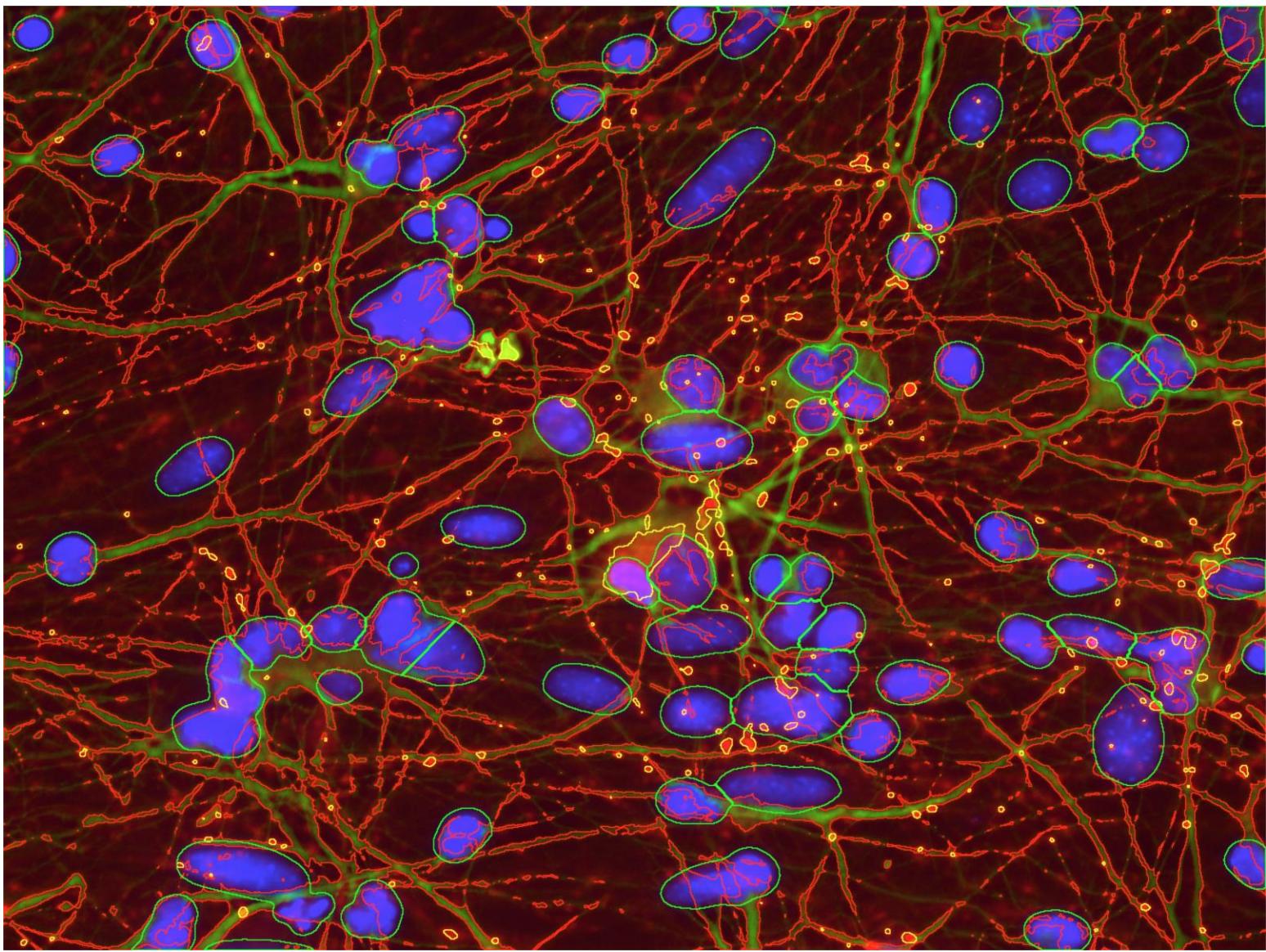
Synapsin Area vs Intensity Distribution



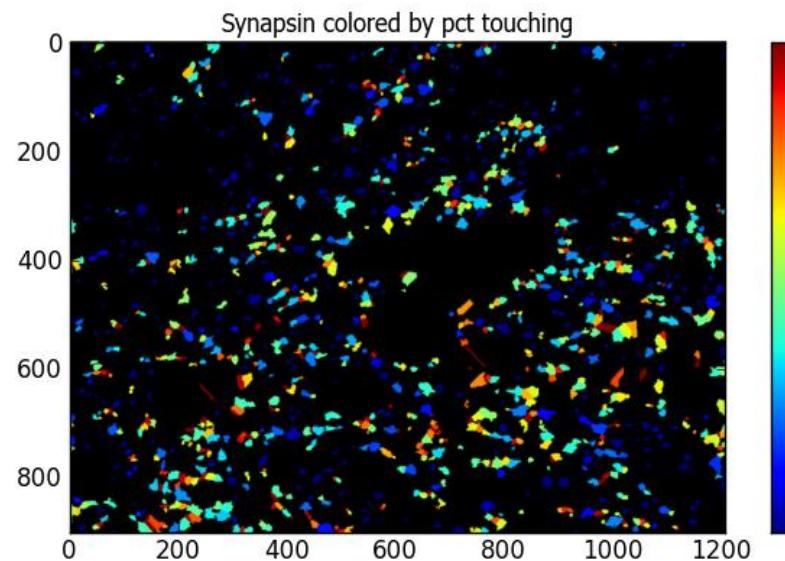
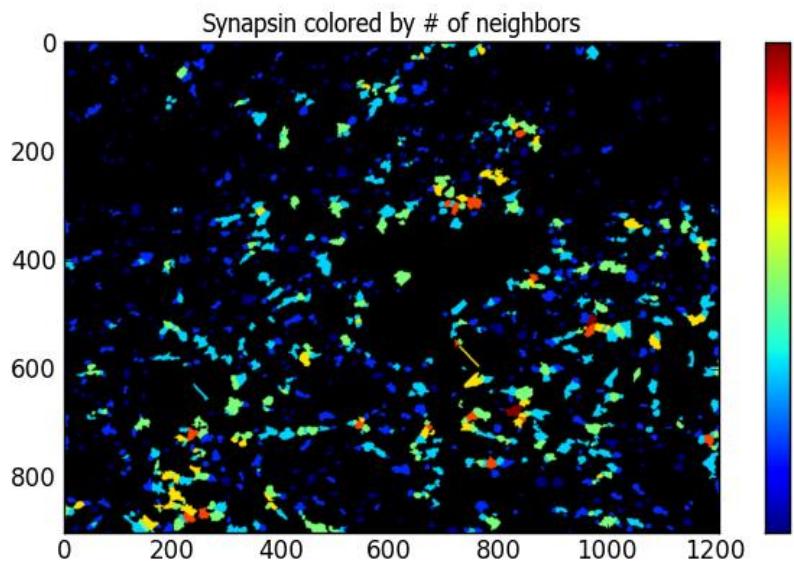
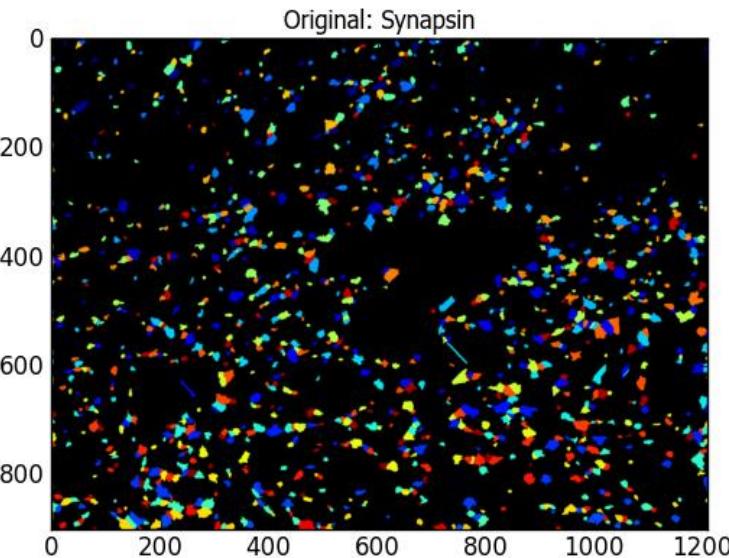
Synapsin objects



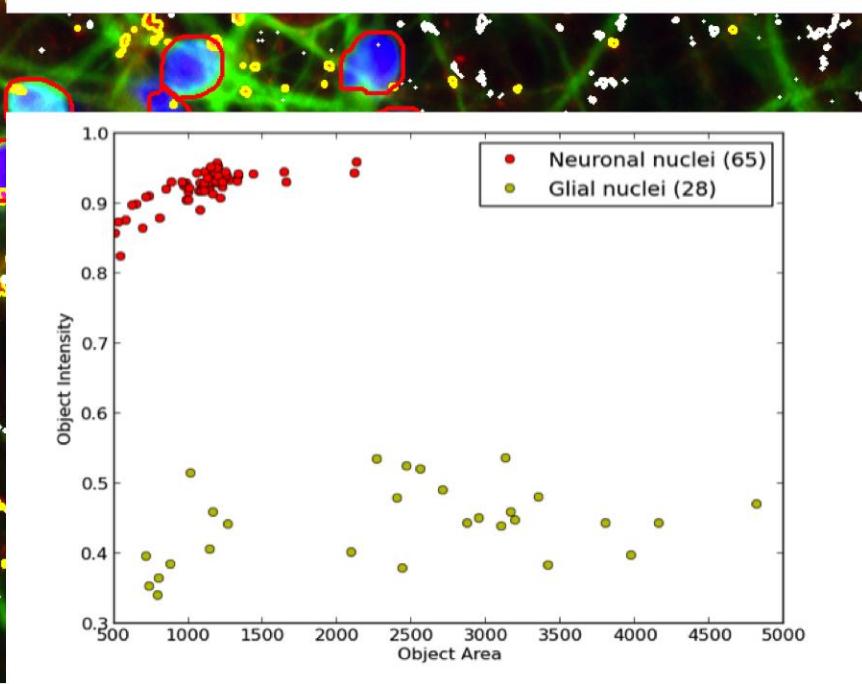
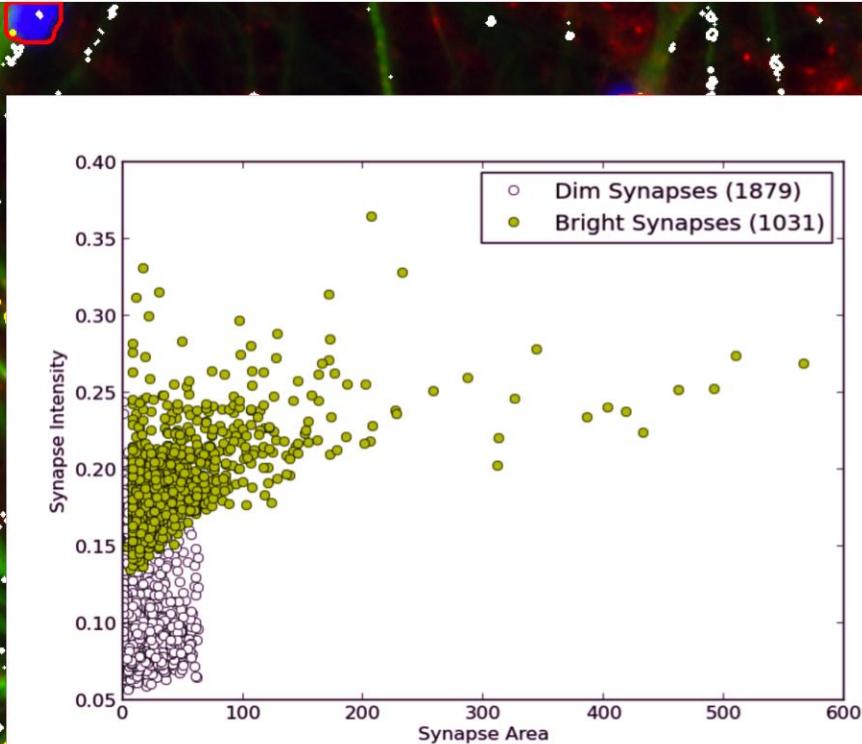
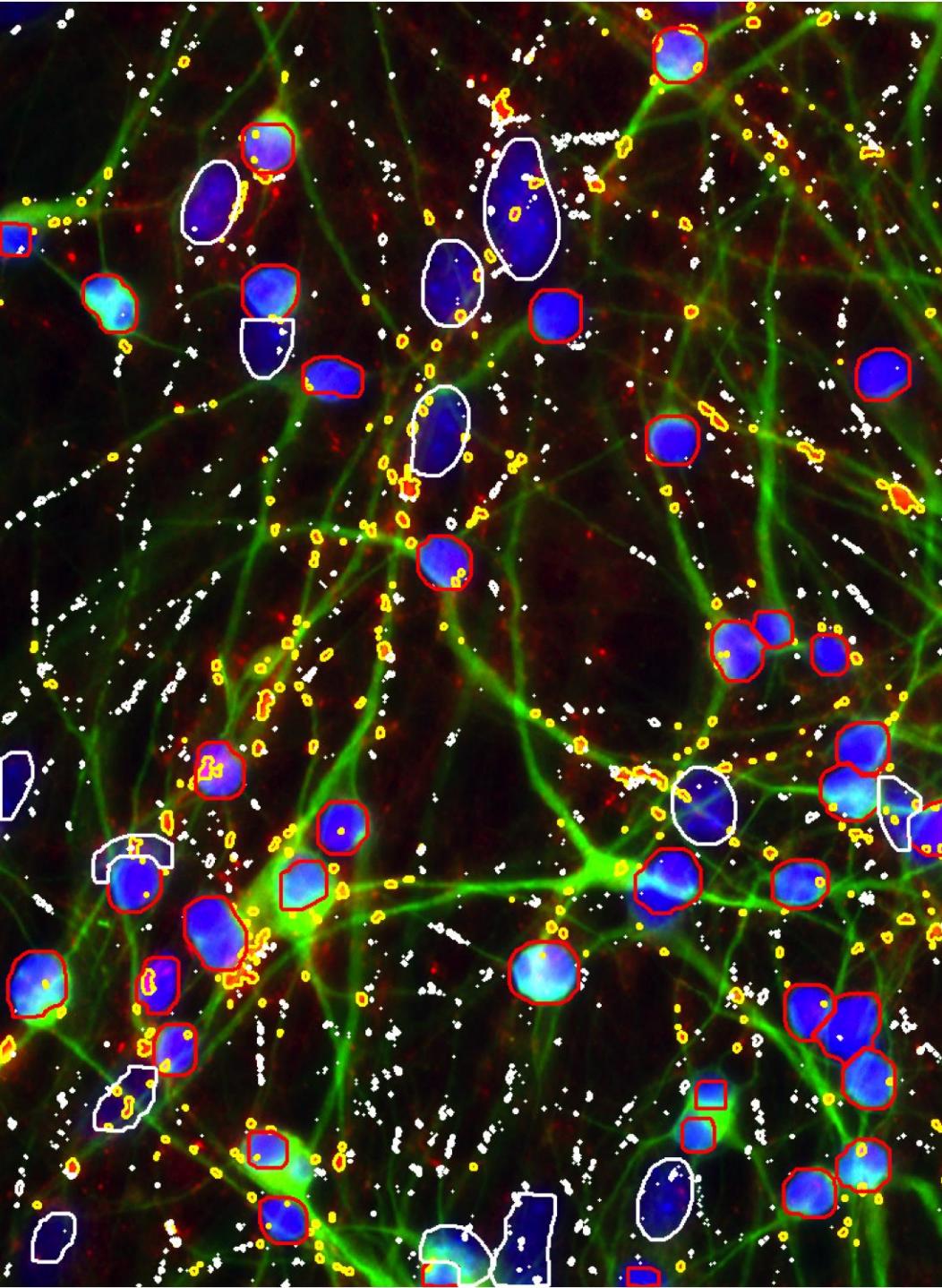
3 Channel Composited/Segmented Image



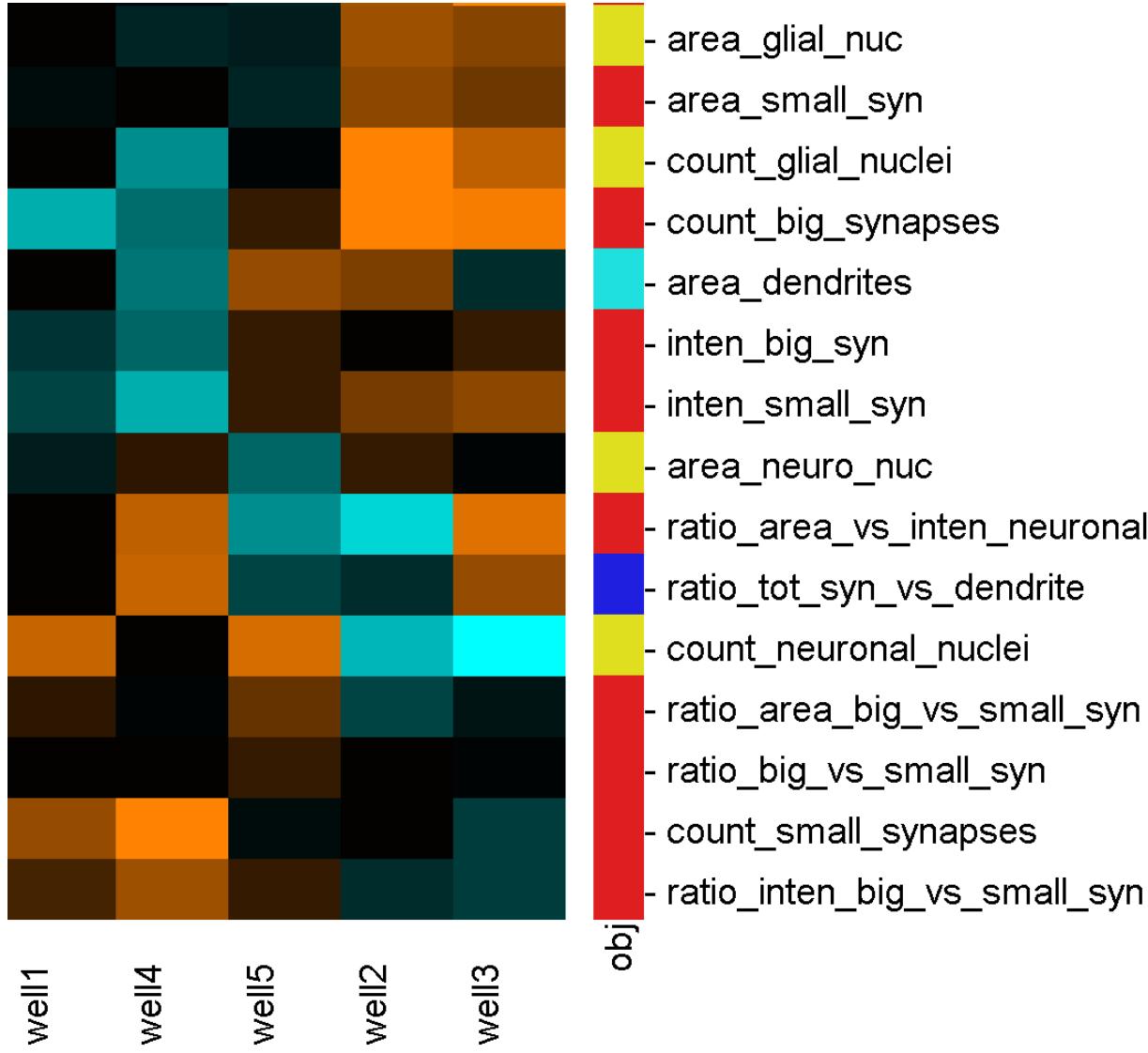
Object Neighbor Count



Counting and quantitating each object per well and generating a feature matrix

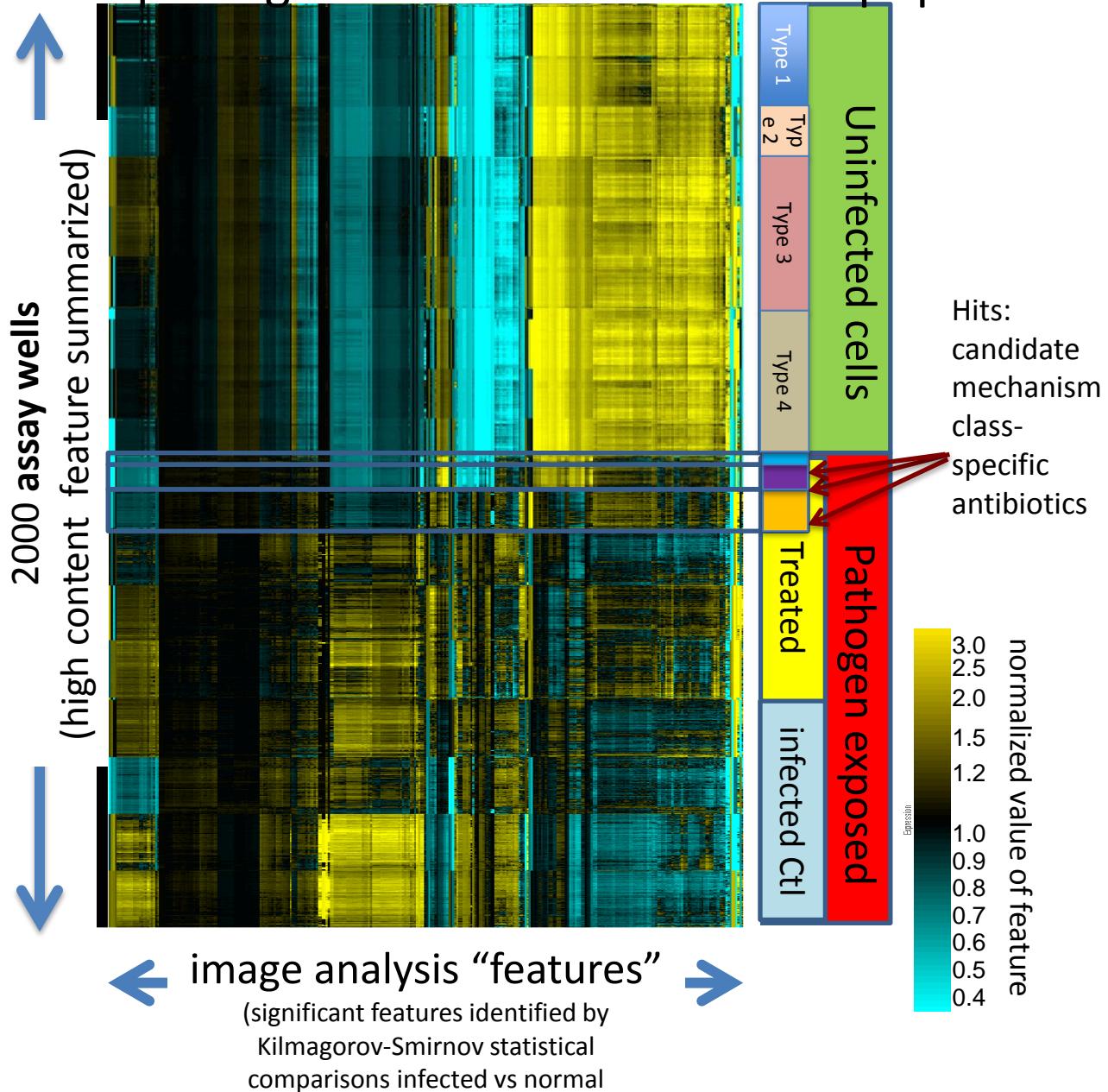


Using quantitative feature matrix to detect informative patterns



Note that the several glial (astrocyte) features (eg # of glial nuclei) are positively correlated with the count of big synapses and inversely correlated to the number of small synapses.

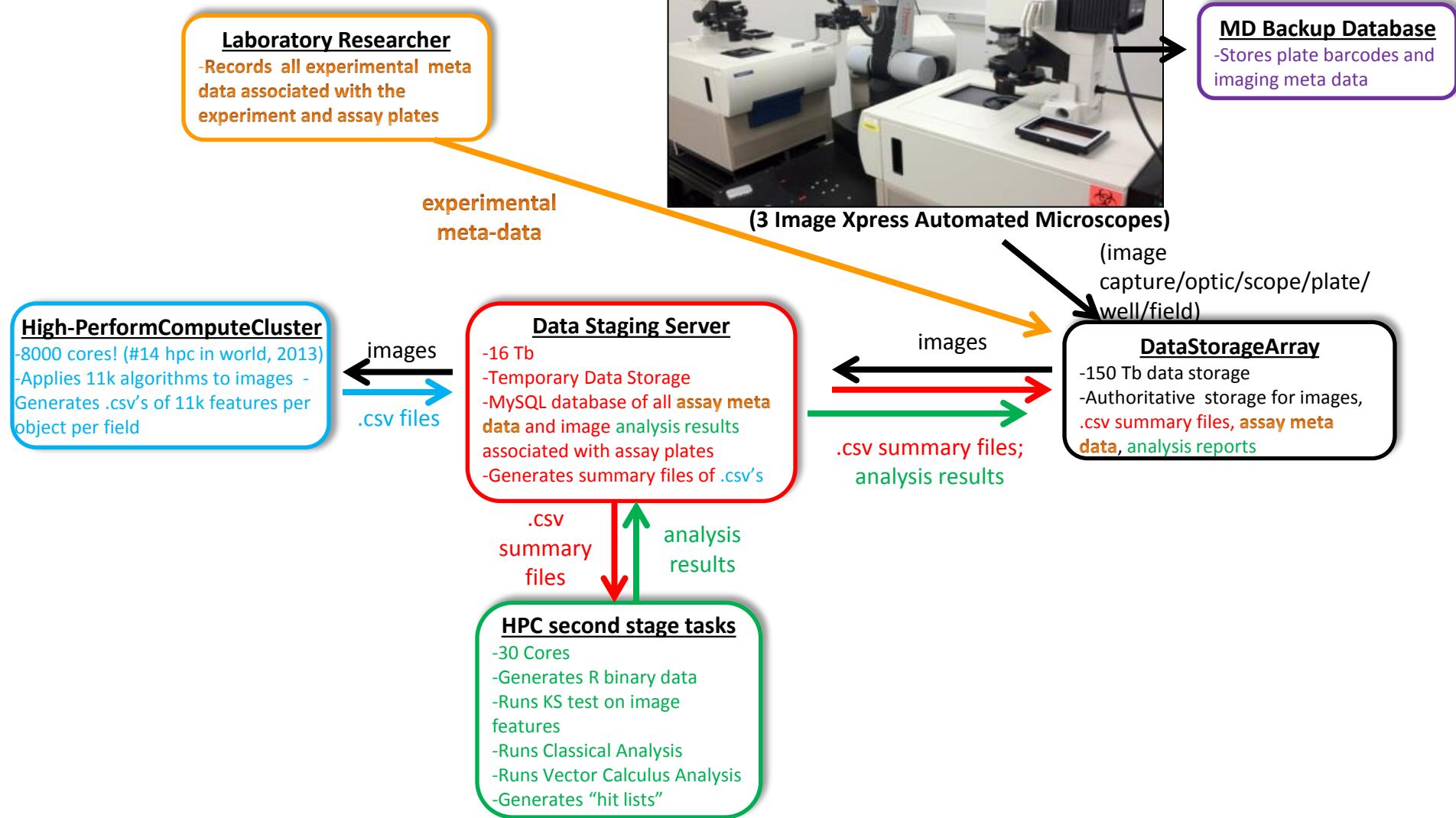
High Content Image Analysis- based profiling of differential morphological characteristics of cell populations in micro-well formats



Based on the population of cells, axons, and synapses in an entire culture well, the measures of each component and all of their variable sizes, shapes, and relationships to each other form a series of quantitative “features” per well. Then these well-associated feature profiles can be compared with one another to identify other wells that exhibit similar or dissimilar characteristics that can then provide information as to method of cell culture, induction, genetic effects, and also the affects of modifying factors such as shRNAs or drugs

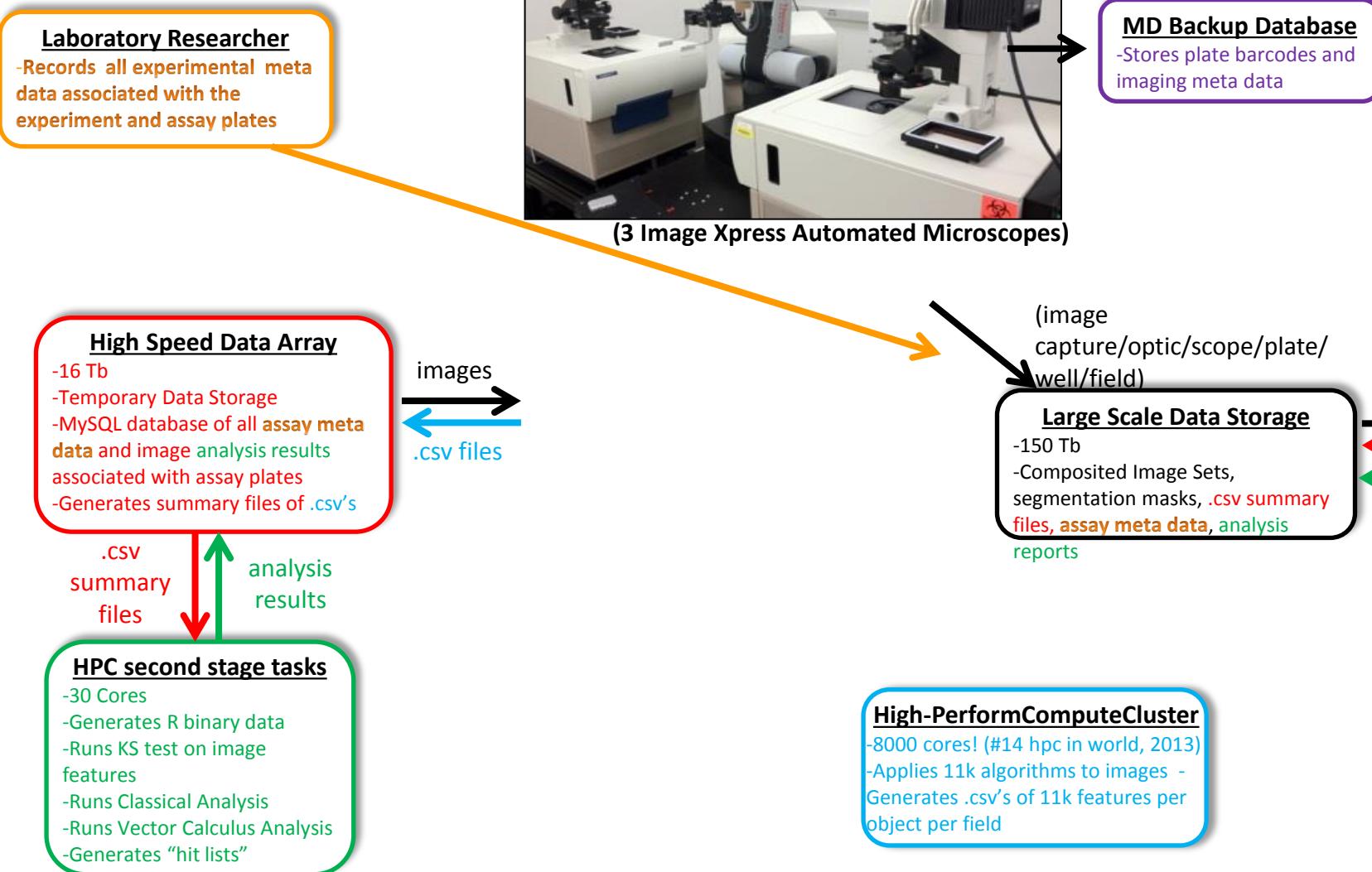
Large Scale Cell Effects-Profiling by High Content Image Capture and Analysis

Hardware, network and data capture pipeline



Large Scale Cell Effects-Profiling by High Content Image Capture and Analysis

Hardware, network and data capture pipeline



Data analysis pipeline

Laboratory Researcher creates EBR file

- Records all experimental meta data associated with the experiment and assay plates



(3 Image Xpress Automated Microscopes)

metadata

images

with plate barcodes
and imaging meta data

Data Staging Server

- 16 Tb
- Temporary Data Storage
- MySQL database of all assay meta data and image analysis results associated with assay plates
- Generates summary files of .csv's

images
.csv files

StorageArray

- 150 Tb data storage
- Authoritative storage for images, .csv summary files, assay meta data, analysis reports

.csv
summary
files

LPC

- 30 Cores
- Generates R binary data
- Runs KS test on image features
- Runs Classical Analysis
- Runs Vector Calculus Analysis
- Generates well scores and "hit lists"

HPC

- 400 cores
- Applies 4K or 11k feature algorithms to images
- Generates .csv's of 11k features

Web Server

- MySQL Database of Analyses, results, reports
- User logins, custom lists for candidate/validated compounds, wells, graphs

Data analyst / researcher

- Examine plates
- Graph and Heatmap of Plate Results
- Examine Well Images
- Select hits reports and analyses results to download by project, batch, plate
- Create/save lists of drugs, shRNA, protocol variables that have impacts