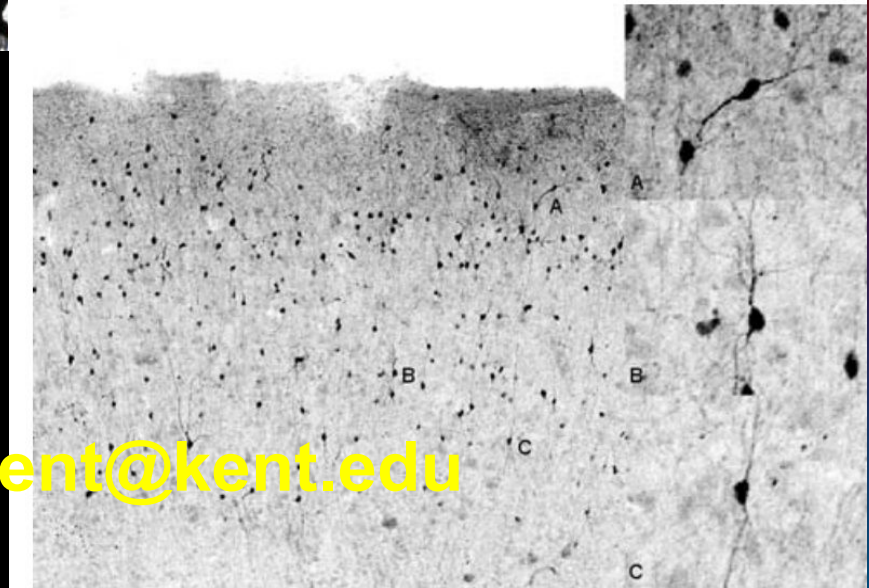
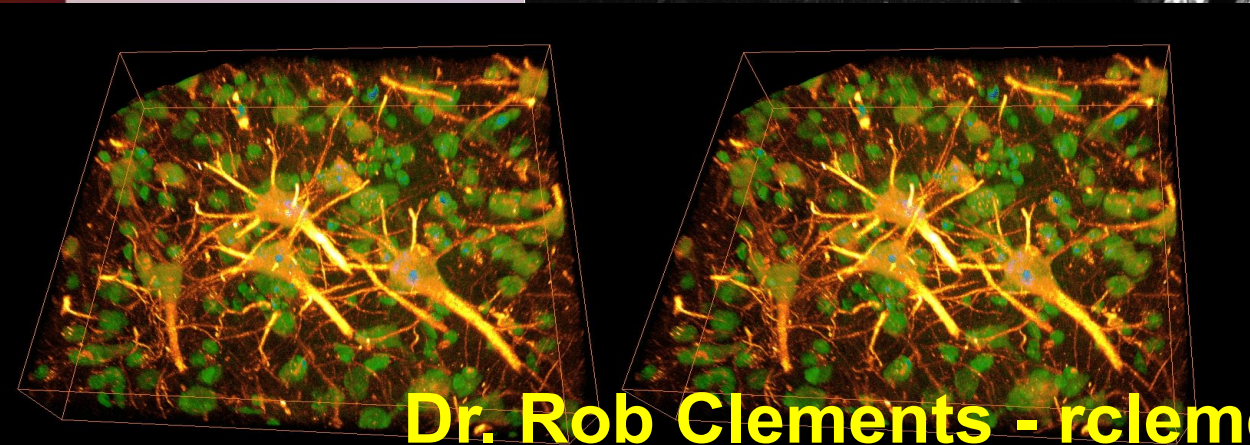
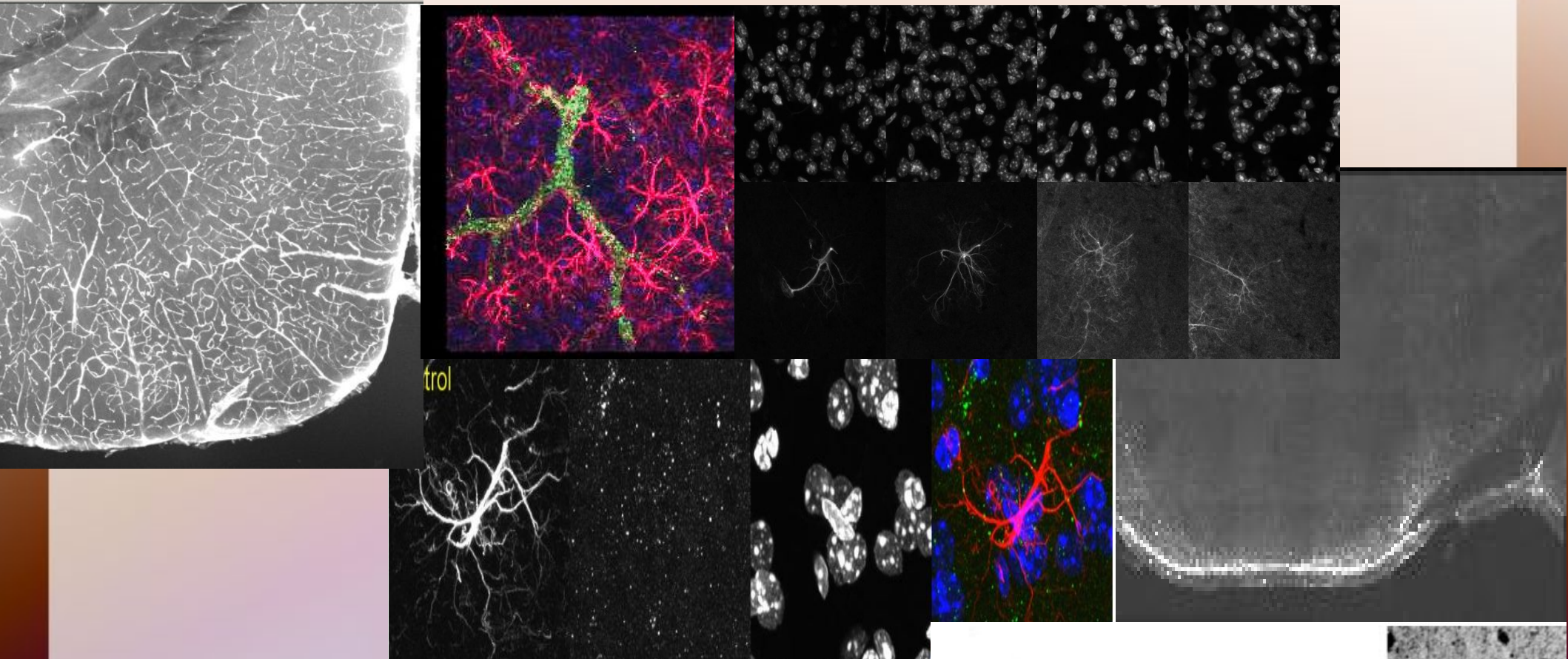


# *Biological Image Analysis*



**Dr. Rob Clements - [rclements@kent.edu](mailto:rclements@kent.edu)**

# Course Overview

- Design methods to automatically analyze microscope images
- Group project based
- Presentations/discussions
- Final products:
  - Presentation
  - **Working** plugin or macro for imagej
  - Report
- .....*lab openings*.....

# Imaging Projects – *Plugins Creation*

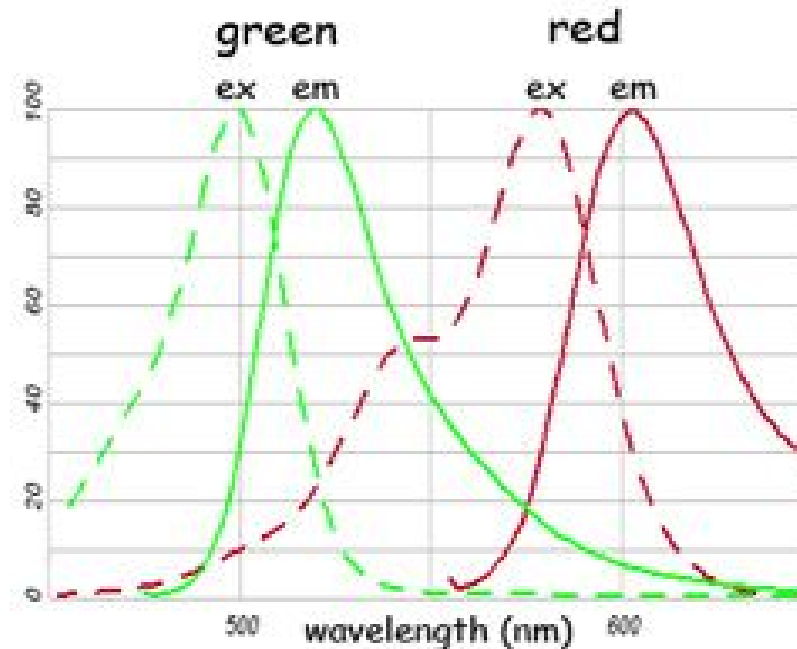
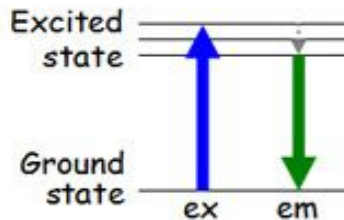
## MICROSCOPY

- 1) Tiled data reconstruction and correction
- 2) Astrocyte Cell classification
- 3) Microvessel analysis



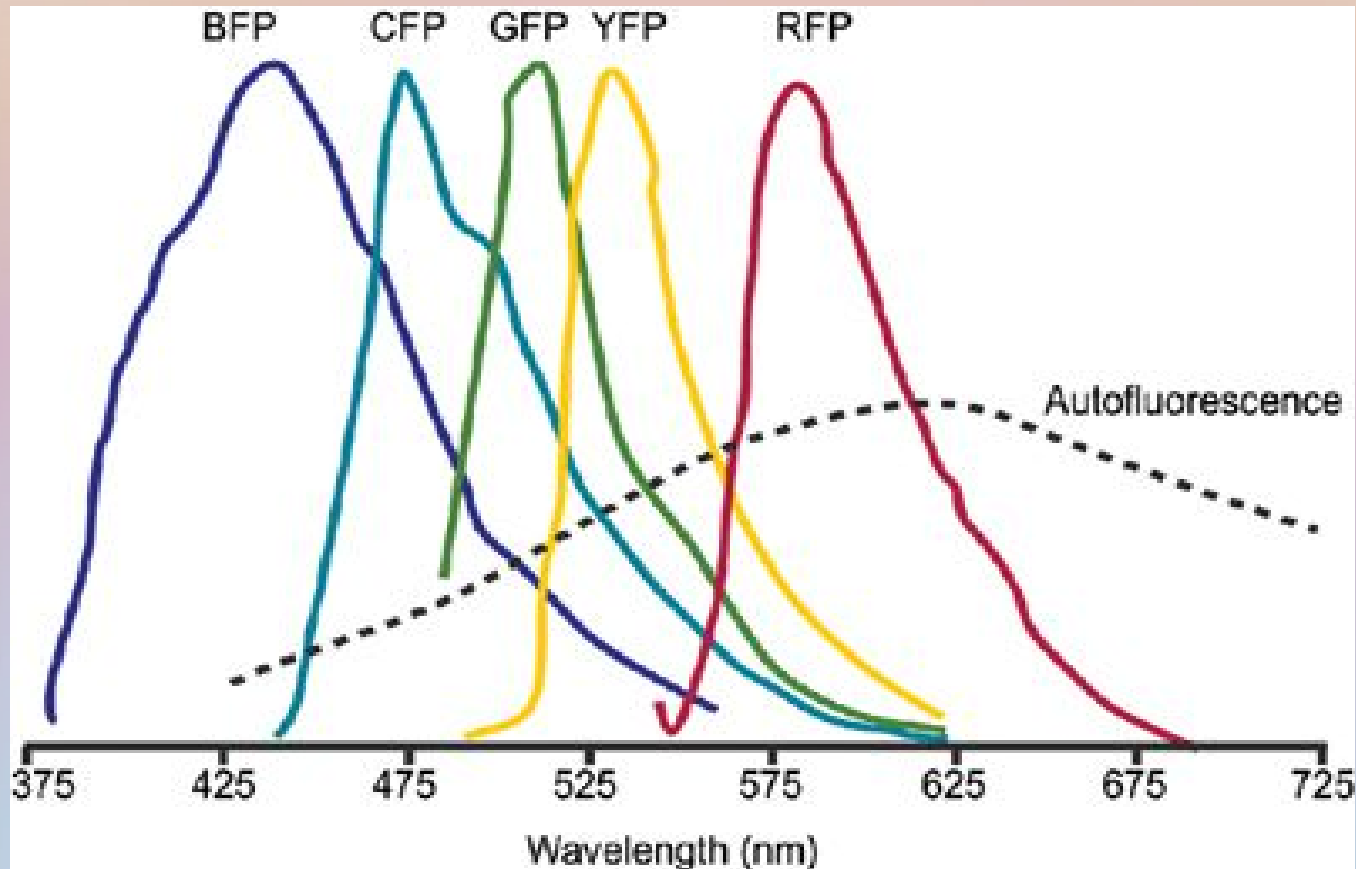
# Fluorescence Microscopy

- specimen can be labeled with a fluorophore (green fluorescent protein-GFP) directed at an object of interest
- specimen is illuminated with light of a specific wavelength which is absorbed by the fluorophores
- fluorophore emits longer wavelength (different color)
- green fluorophore absorbs a blue excitation photon and an electron changes from the ground state to an excited state.
- electron then decays back from the excited state to the ground state and releases a green photon

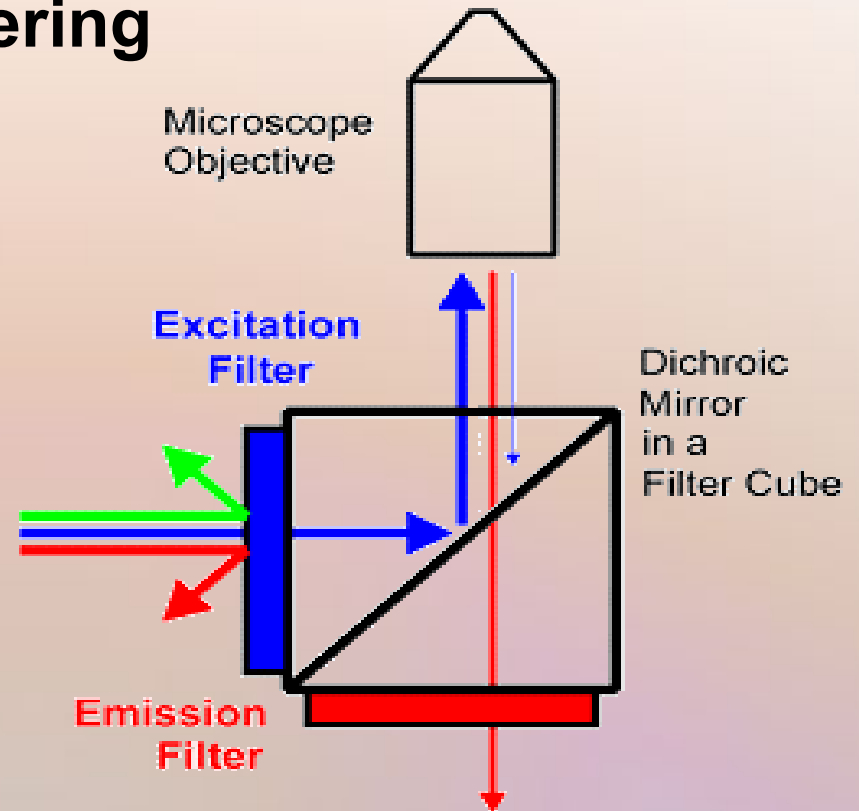
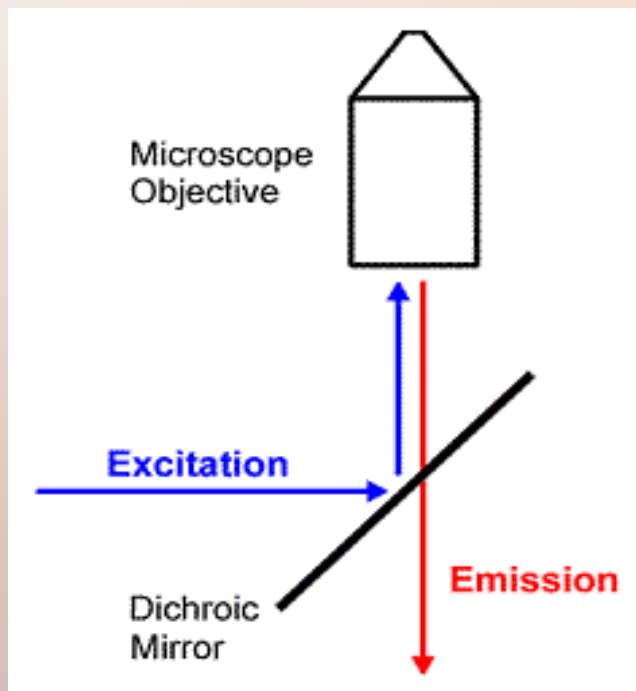


# Fluorescence Microscopy

- Possible to use concurrent fluorophores as long as excitation/emission wavelength different
- commercially available bound to organelle probes and antibodies
- permits probing multiple structures concurrently



# Fluorescence Microscopy Wavelength Filtering

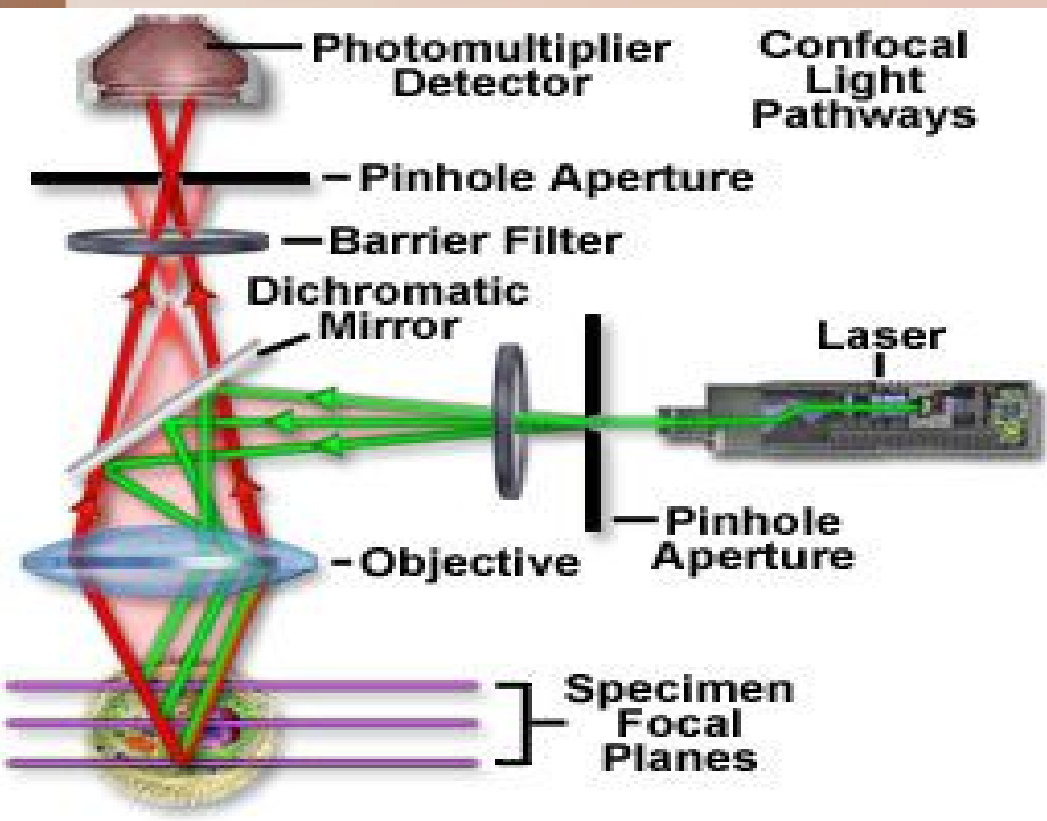


\* **Excitation filter** - the excitation path just prior to the dichroic mirror to select the excitation wavelength

\* **Emission filter** - to select the emission wavelength from the sample and remove excitation light, it is placed beneath the dichroic mirror.

# Confocal Microscopy

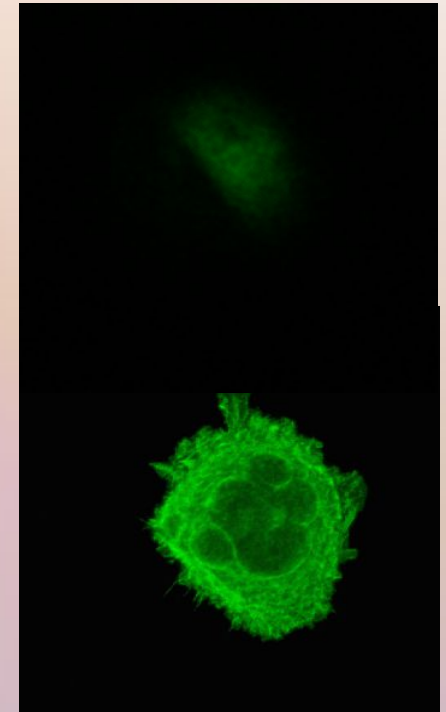
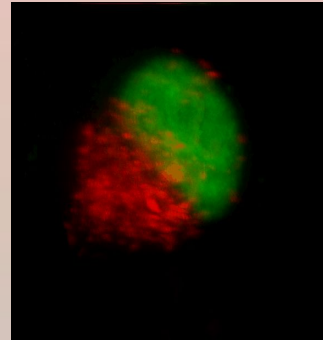
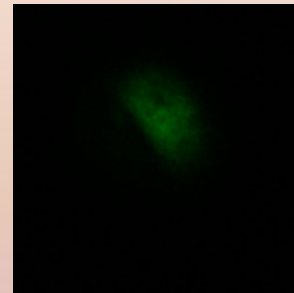
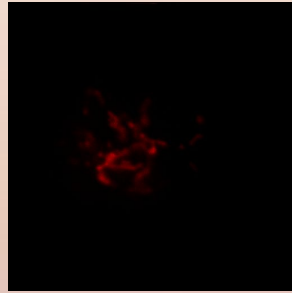
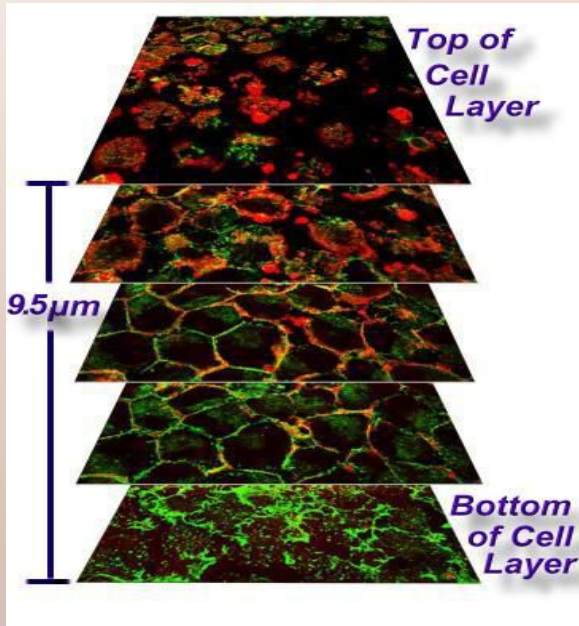
- laser line to illuminate and scan samples, with series of mirrors and filters
- uses point illumination and pinhole to eliminate out-of-focus light





# Confocal Microscopy

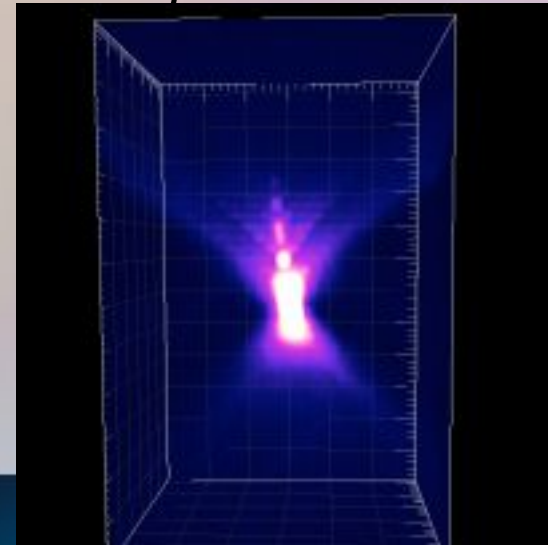
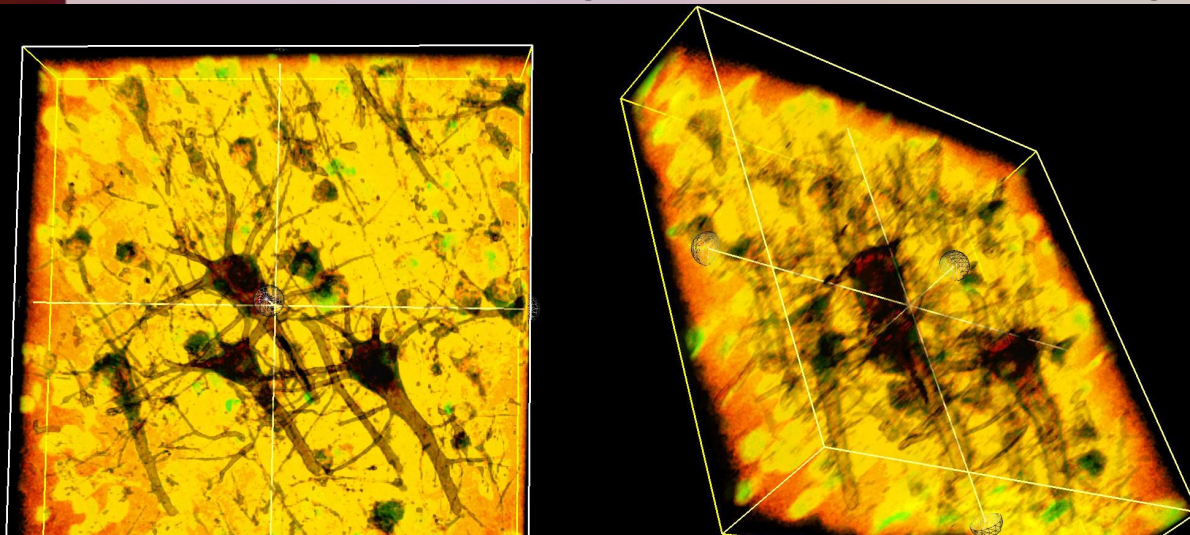
Permits 3D data collection by creating image stacks and 3D reconstruction  
-far better resolution by imaging single planes and recombining



## Problems

*Acquisition time, brightness, Axial Blurring*

*Point Spread Function*





# Imaging Projects – *Plugins Creation*

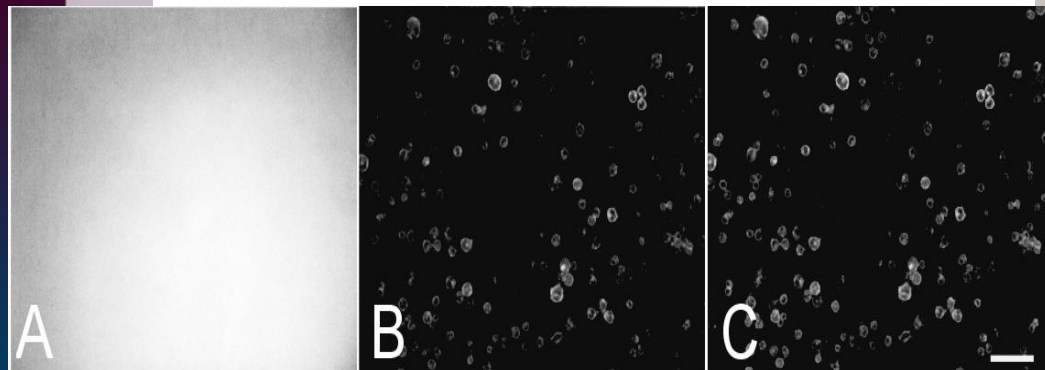
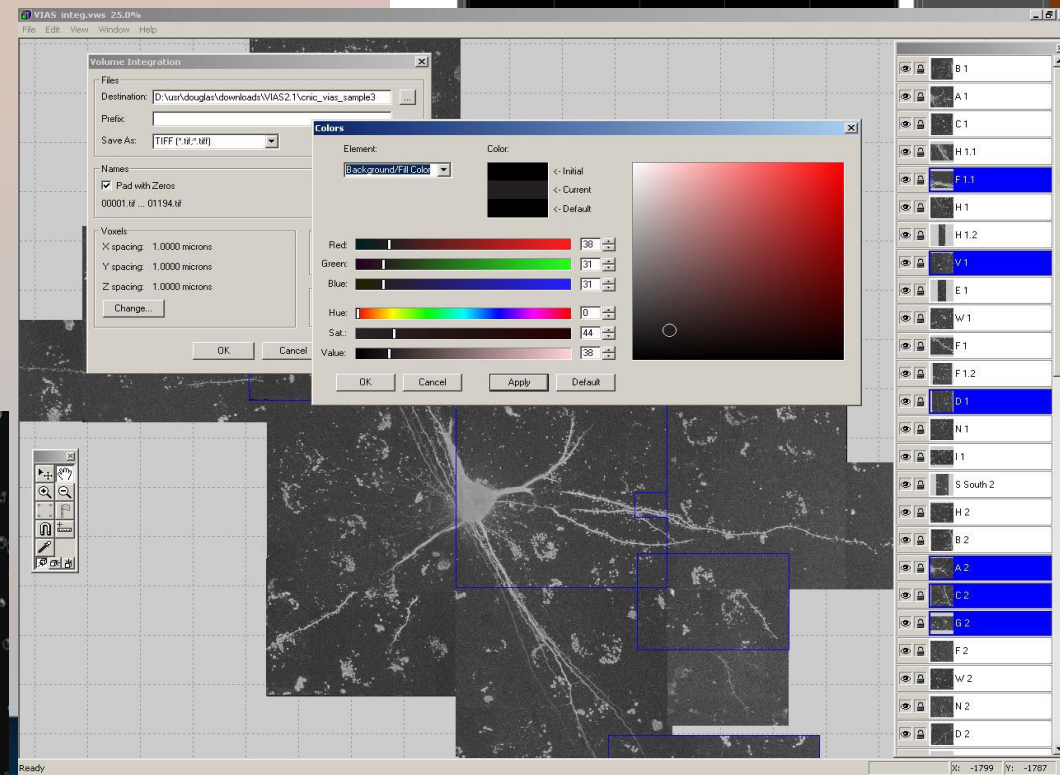
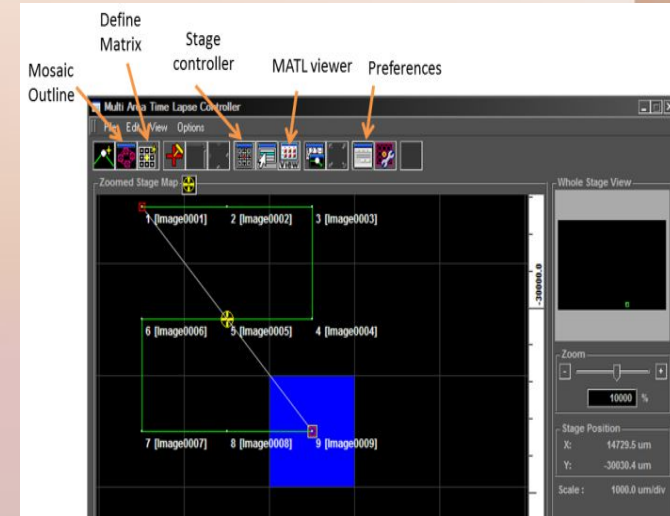
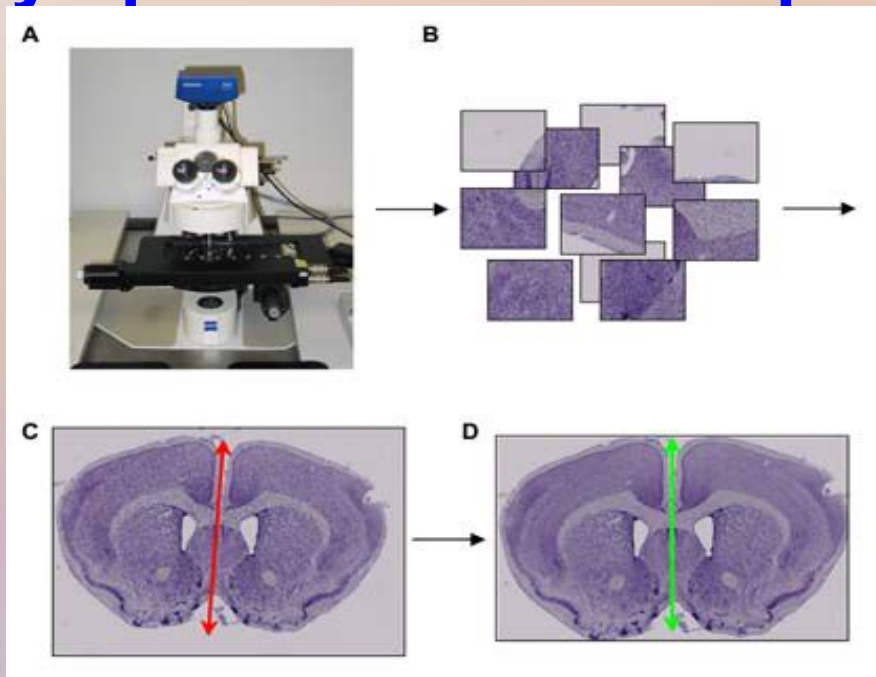
## MICROSCOPY

- 1) *Tiled data reconstruction and correction*
- 2) **Astrocyte Cell classification**
- 3) **Microvessel analysis**

# Imaging Projects

## 1) *Tiled data reconstruction and correction*

- accurately* stitch multiple images together
- correct brightness across images
- existing plugin usage?
- olympus data format input -OIB



# Imaging Projects

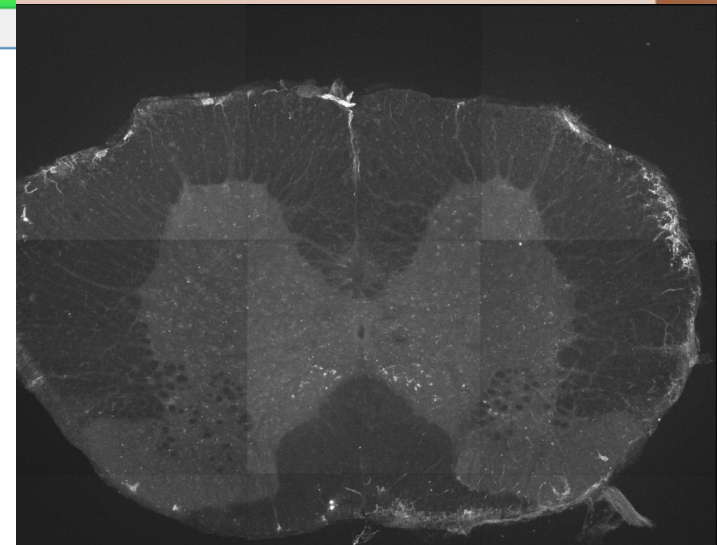
## 1) *Tiled data reconstruction and correction*

-accurately stitch multiple images together

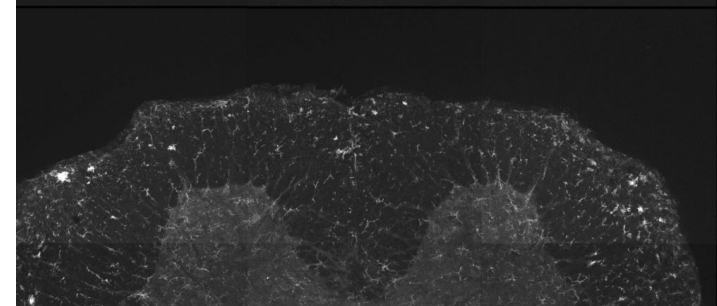
-multichannel OIB and MATL input – rgbTiff/Select output

-stage drift-->predictable?

```
MAI1_Mosaic - Notepad
File Edit Format View Help
/Stage>
<XAxisDirection>LeftToRight</XAxisDirection>
<YAxisDirection>TopToBottom</YAxisDirection>
<NumberOfMosaics>3</NumberOfMosaics>
<Repeats>1</Repeats>
<Version>1.1</Version>
<ProjectionFolder Path="Relative">\Projection</ProjectionFolder>
<AppendFolder Path="Relative">\Append</AppendFolder>
<AppendProjectionFolder Path="Relative">\AppendProjection</AppendProjectionFolder><Mosaic
="3">
  <Stitch>True</Stitch>
  <XScanDirection>LeftToRight</XScanDirection>
  <YScanDirection>TopToBottom</YScanDirection>
  <XIndex>635.90</XIndex>
  <YIndex>635.90</YIndex>
  <IndexRatio>100</IndexRatio>
  <XImages>3</XImages>
  <YImages>3</YImages>
  <ImageInfo>
    <No>1</No>
    <XPos>6724.70</XPos>
    <YPos>-8455.70</YPos>
    <Xno>0</Xno>
    <Yno>0</Yno>
    <Status>True</Status>
    <Filename>Image0001.oib</Filename>
  </ImageInfo>
  <ImageInfo>
    <No>2</No>
    <XPos>7360.60</XPos>
    <YPos>-8455.70</YPos>
    <Xno>1</Xno>
    <Yno>0</Yno>
    <Status>True</Status>
    <Filename>Image0002.oib</Filename>
  </ImageInfo>
  <ImageInfo>
    <No>3</No>
    <XPos>7996.50</XPos>
    <YPos>-8455.70</YPos>
    <Xno>2</Xno>
    <Yno>0</Yno>
    <Status>True</Status>
    <Filename>Image0003.oib</Filename>
  </ImageInfo>
</Mosaic>
</Stage>
```



Collagen IV (vessels)

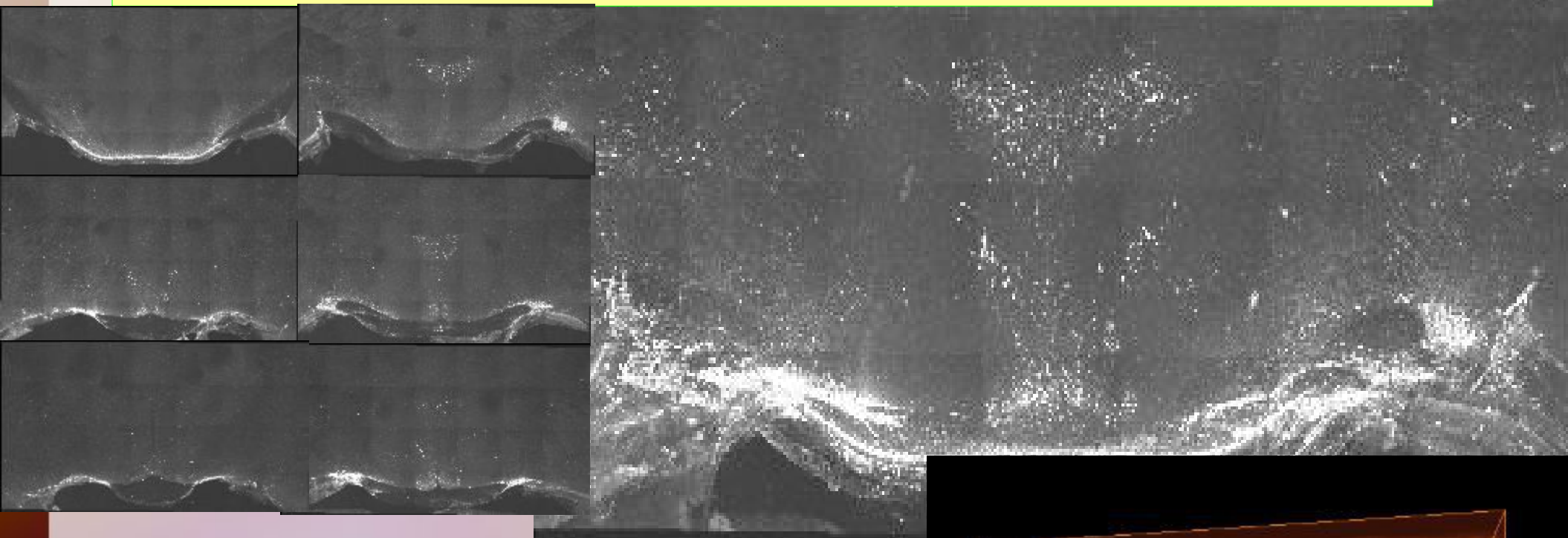




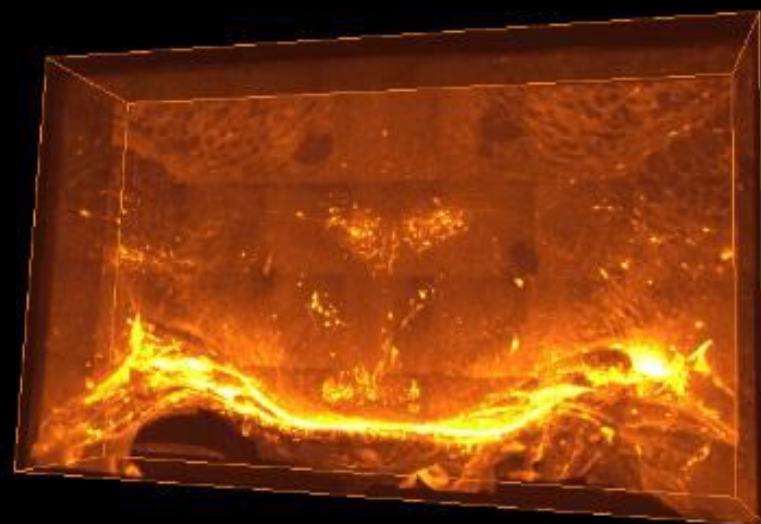
# Imaging Projects

## 2) Tiled data reconstruction and correction

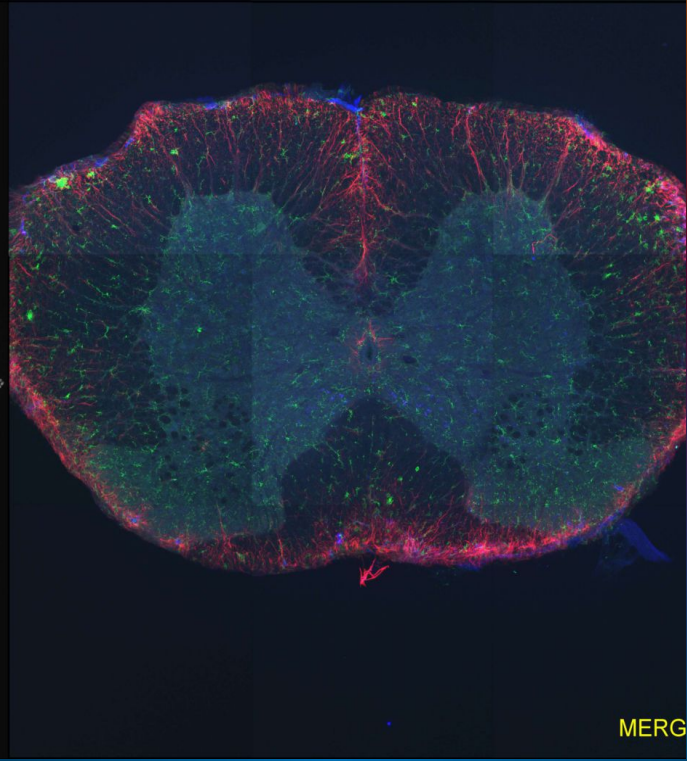
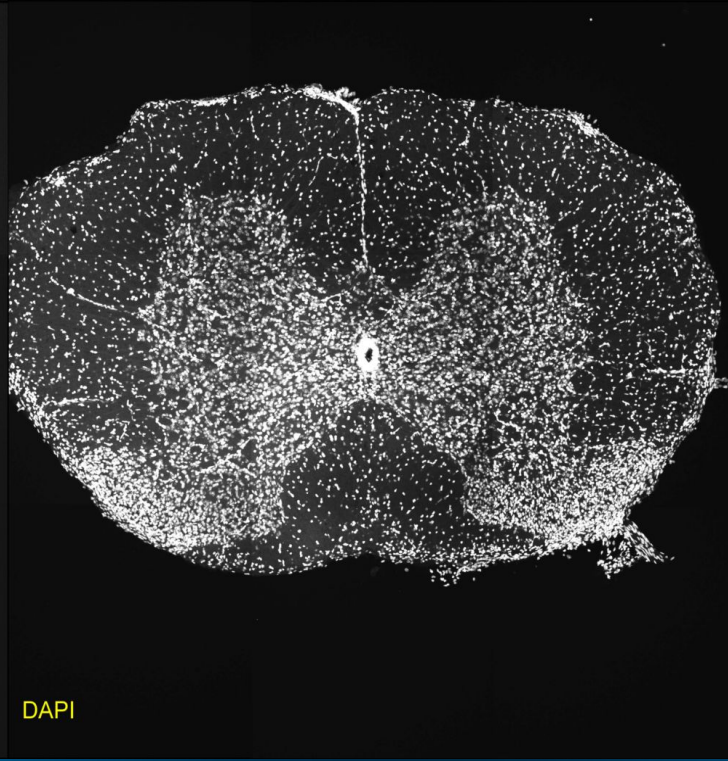
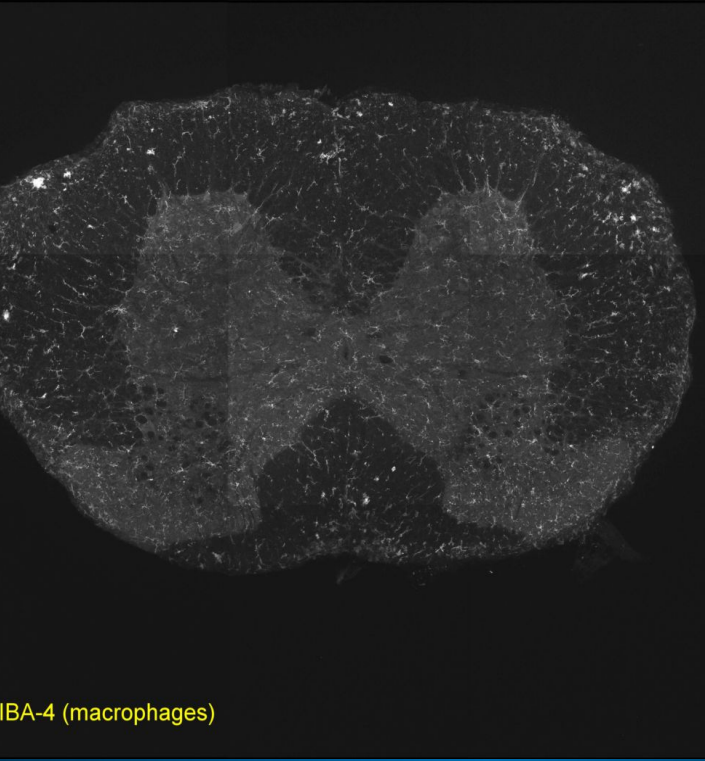
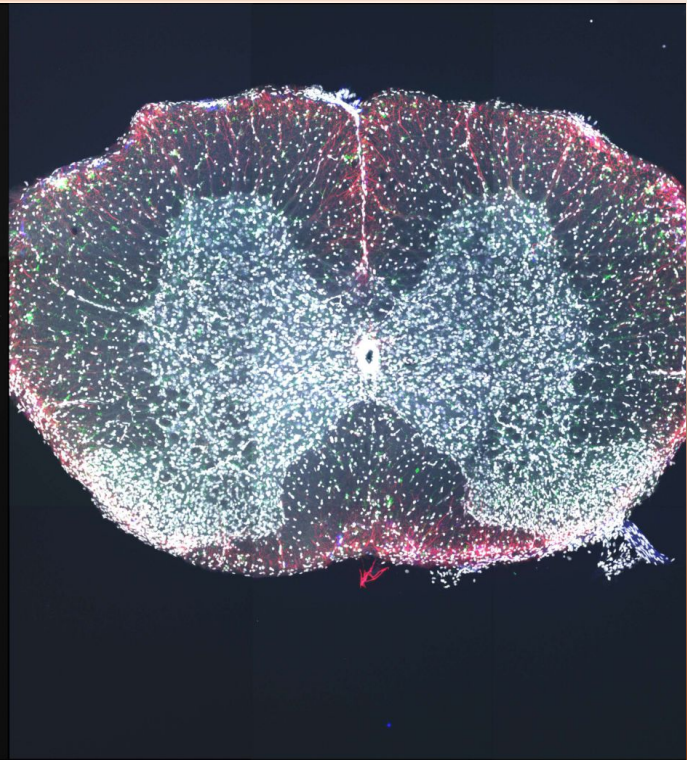
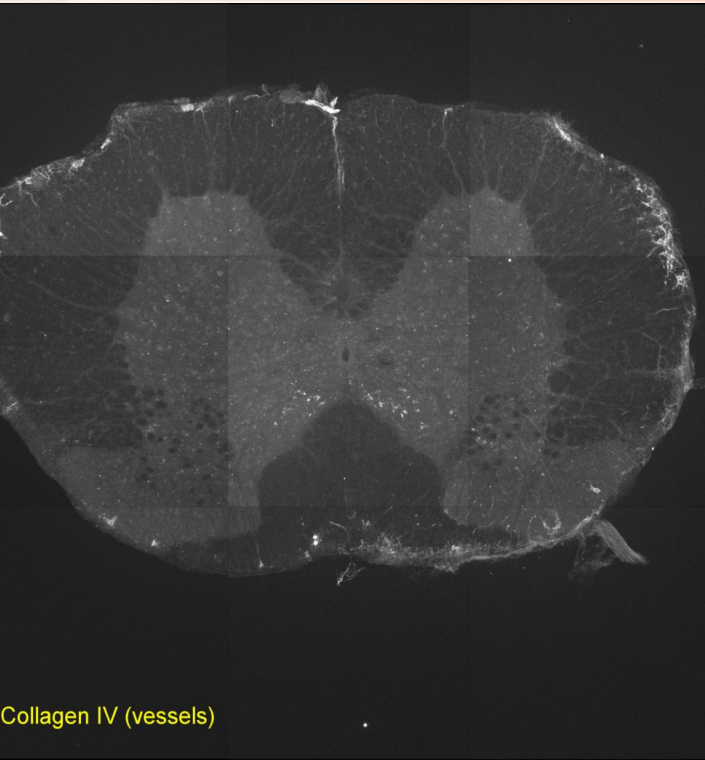
### TISSUE RECONSTRUCTION



- 4.35 mm x 2.6mm x 1.4 mm
- voxel resolution; 1.2 microns
- 3508x2072x700 pixels
- 19600 optical sections





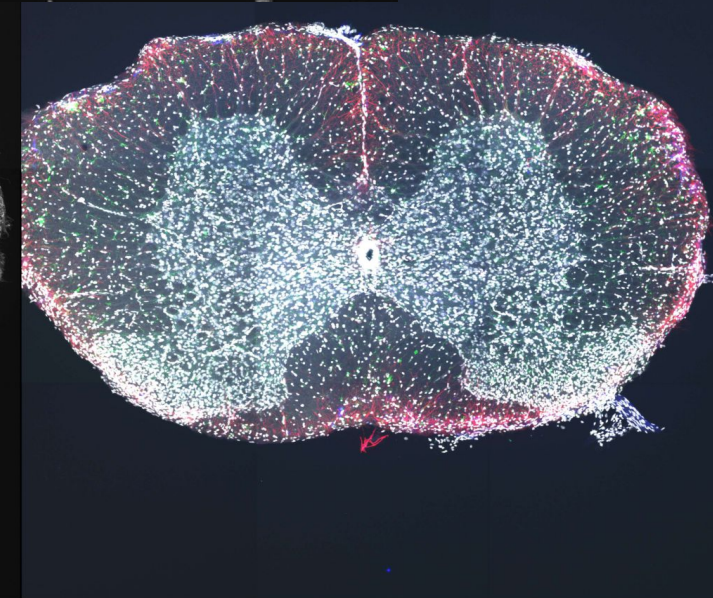
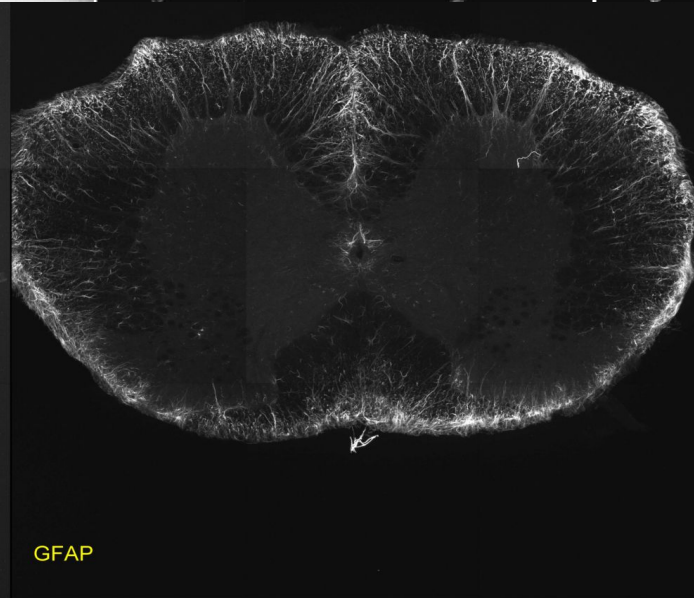
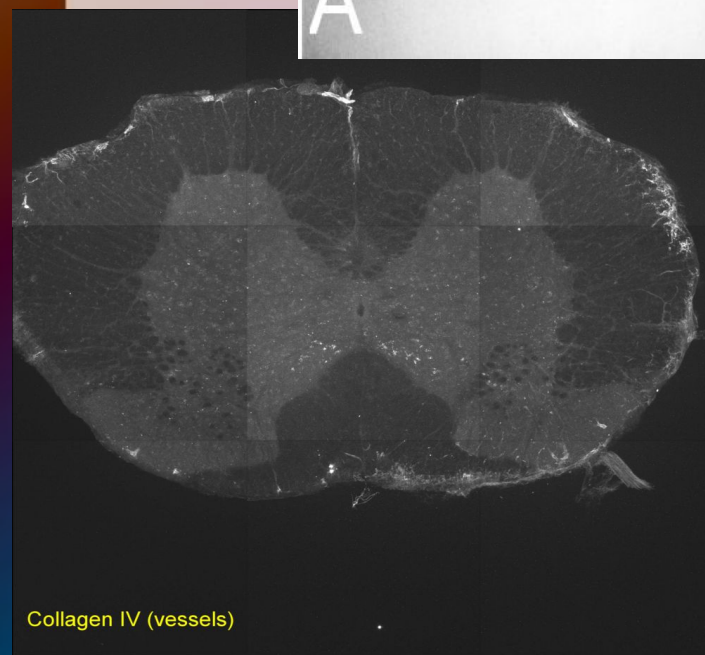
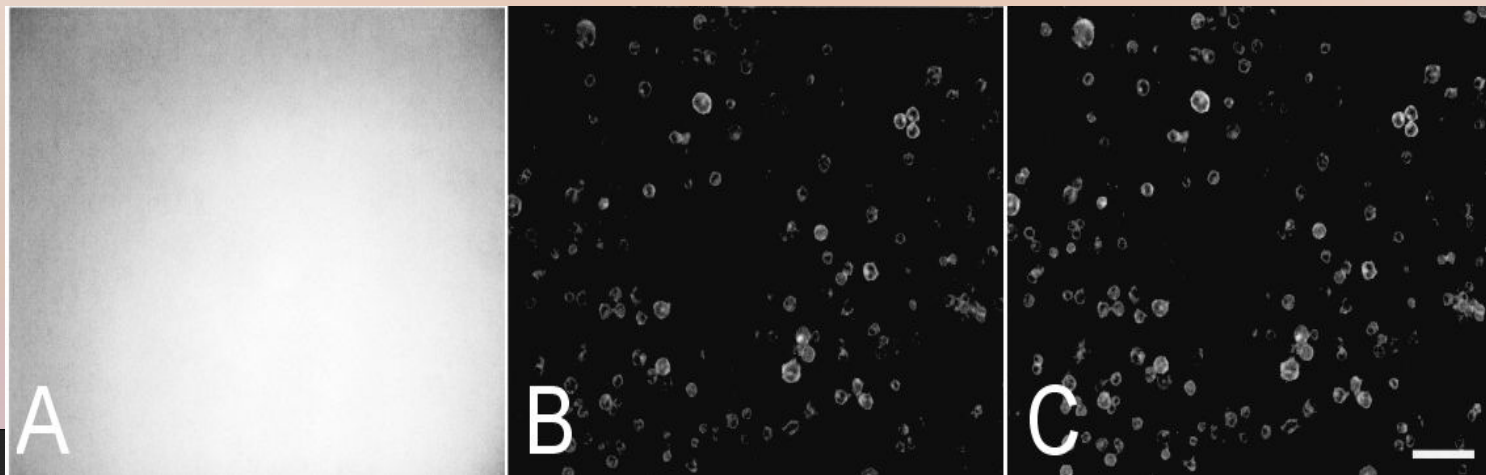




# Imaging Projects

## 2) *Tiled data reconstruction and correction*

- correct brightness across images/seems
- use adjacent tiles, (calibration image?)
- different channel issues?



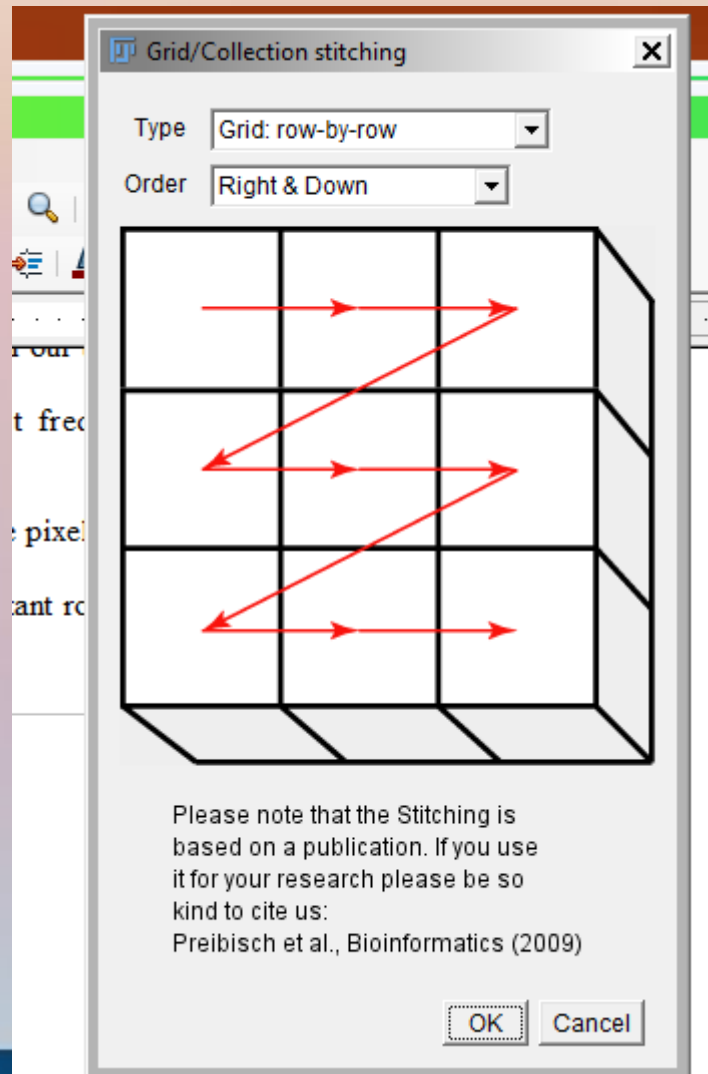


# Existing Tools

*-grid collection stitching plugin*

[https://imagej.net/Grid/Collection\\_Stitching\\_Plugin](https://imagej.net/Grid/Collection_Stitching_Plugin)

(no brightness correction, need to read MATL file)



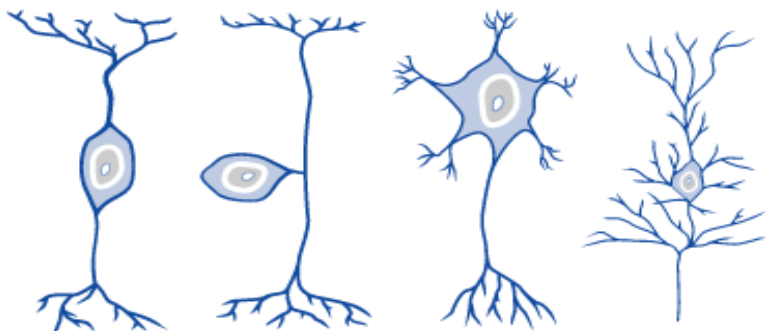
# Imaging Projects – *Plugins Creation*

## MICROSCOPY

- 1) Tiled data reconstruction and correction
- 2) *Multiple Astrocyte Cell classification*
- 3) Microvessel analysis

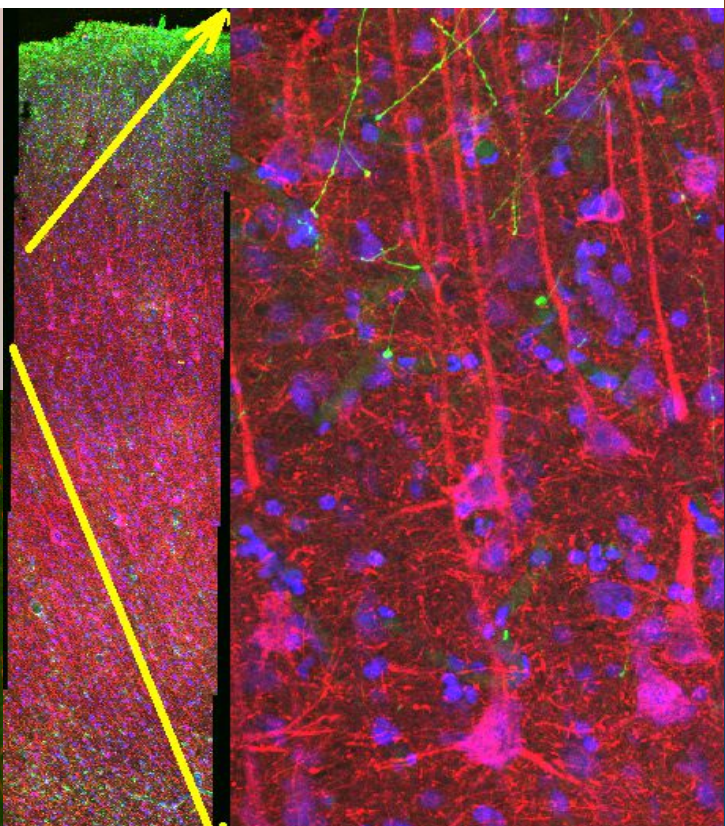
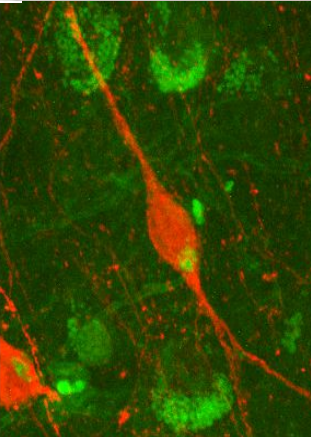
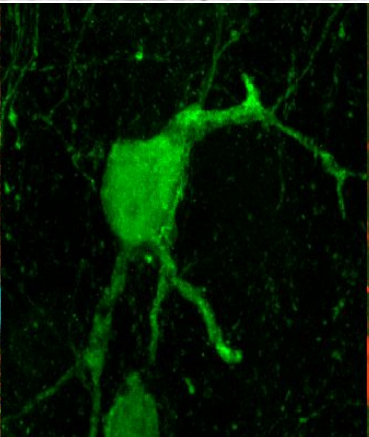
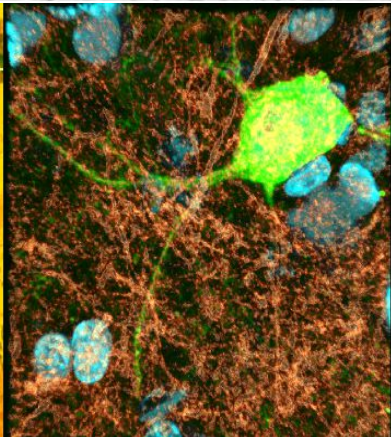
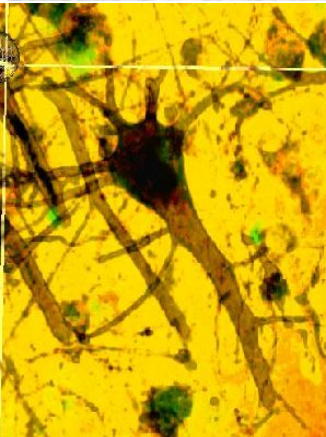
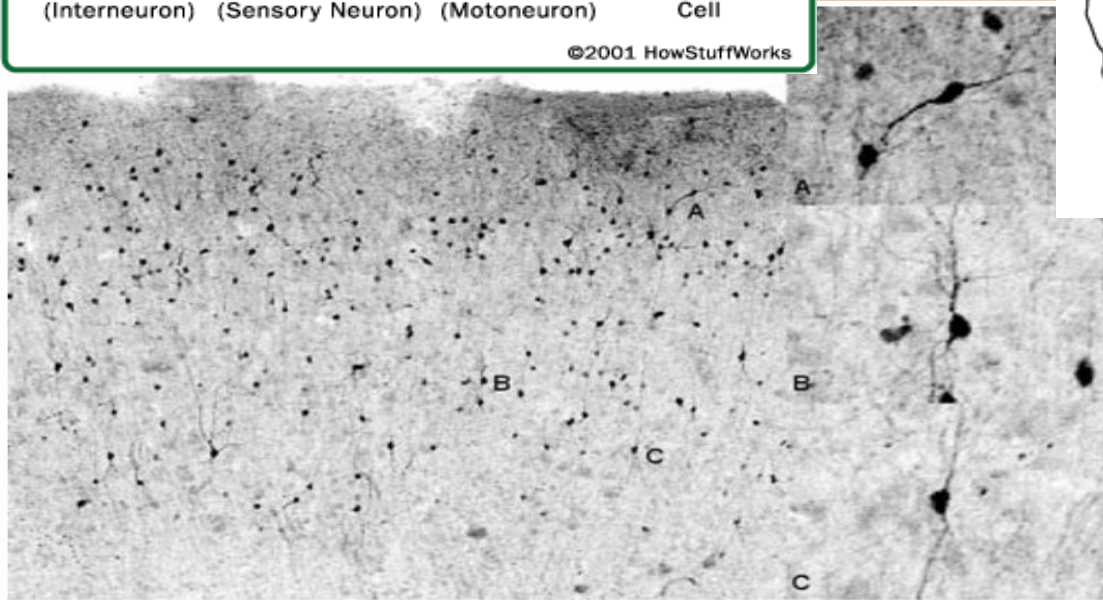
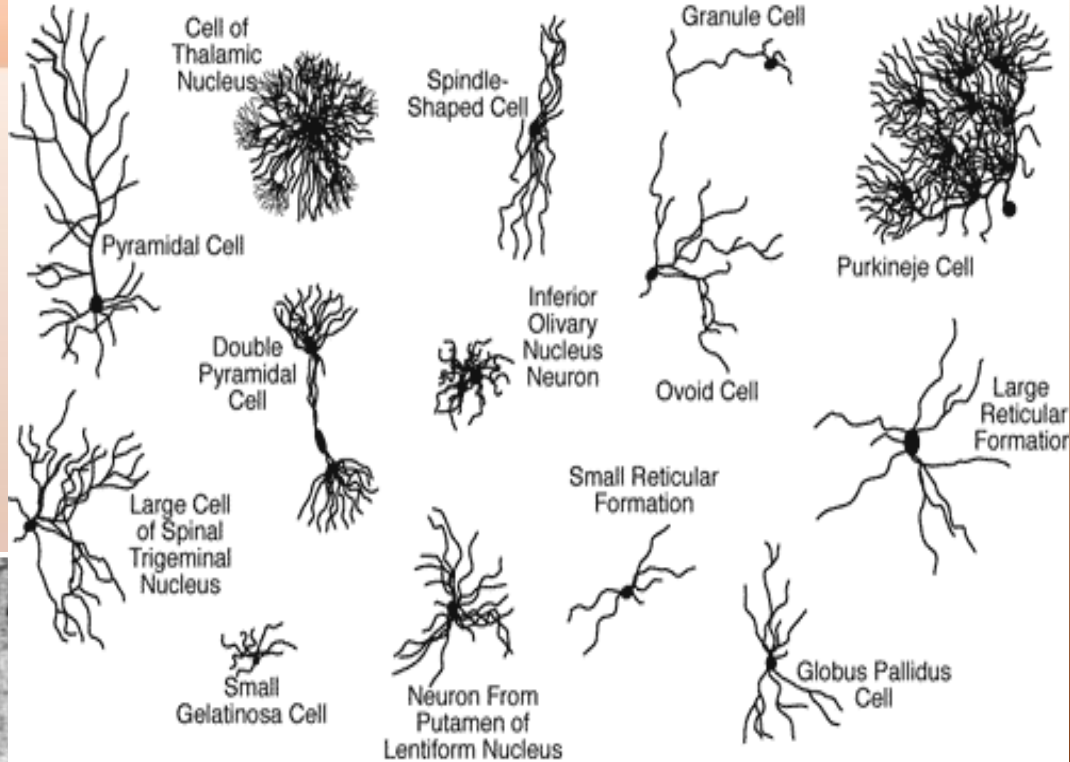
# Types of Neurons

## Basic Neuron Types



Bipolar (Interneuron)    Unipolar (Sensory Neuron)    Multipolar (Motoneuron)    Pyramidal Cell

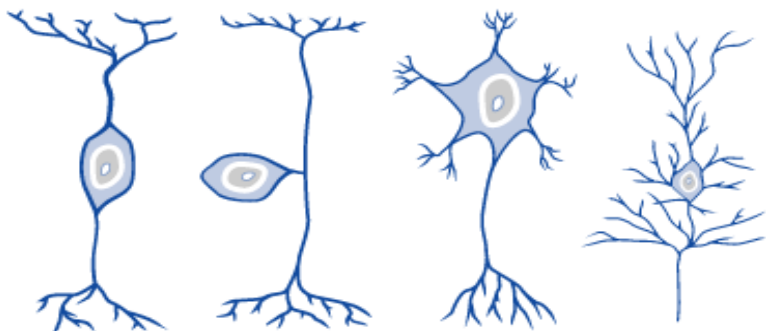
©2001 HowStuffWorks





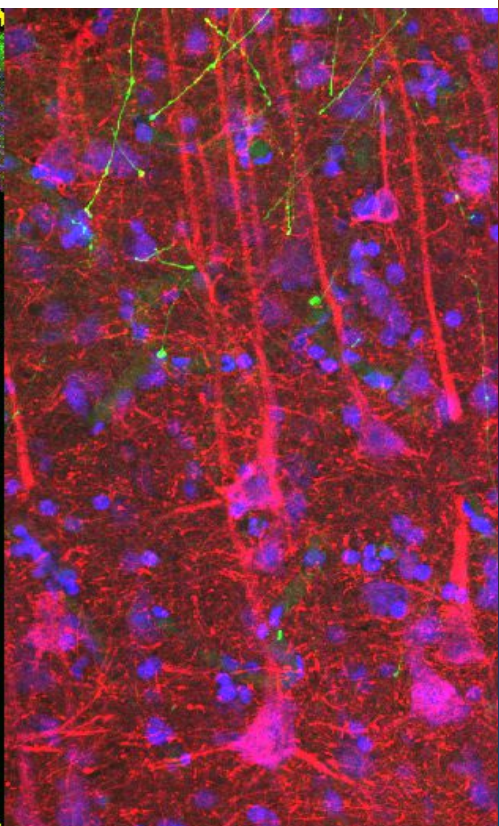
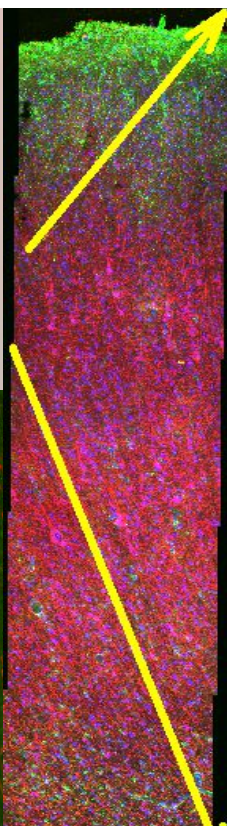
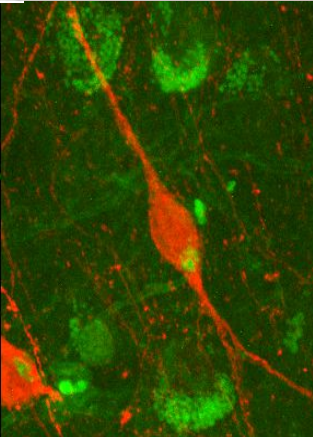
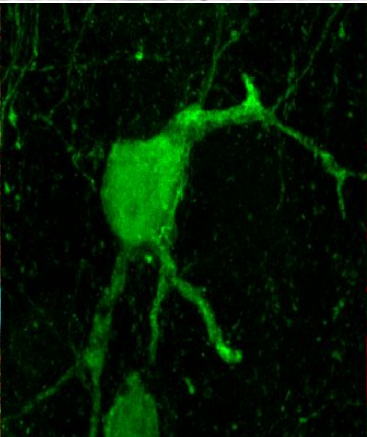
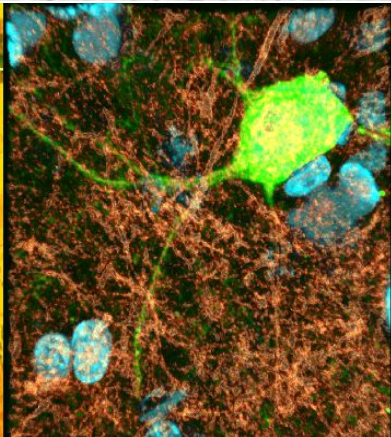
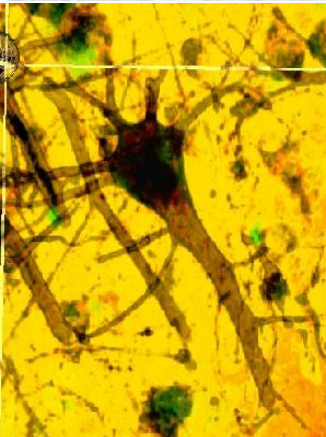
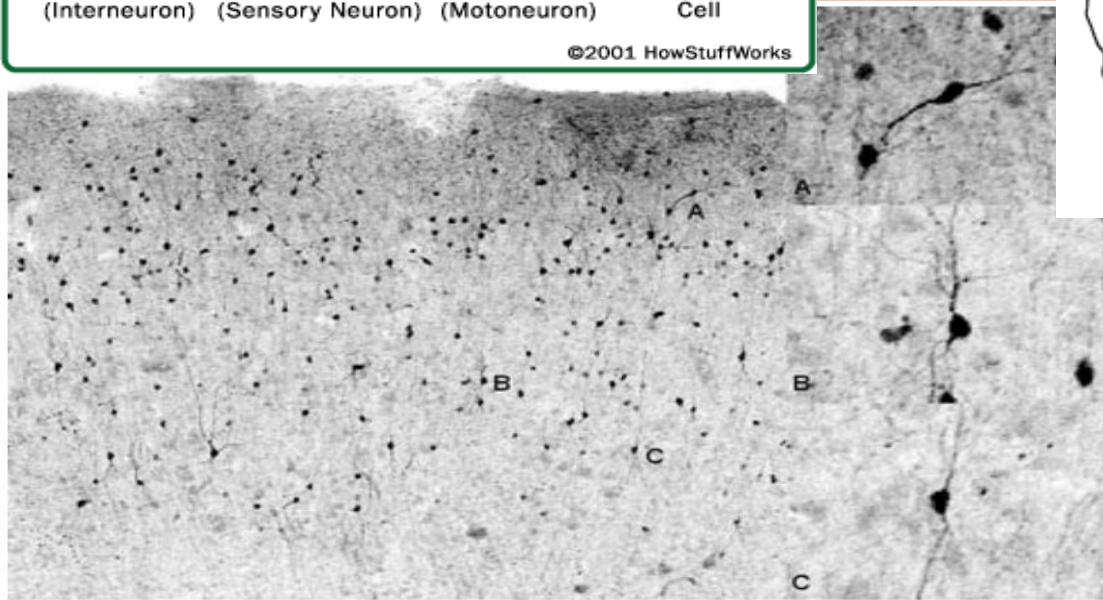
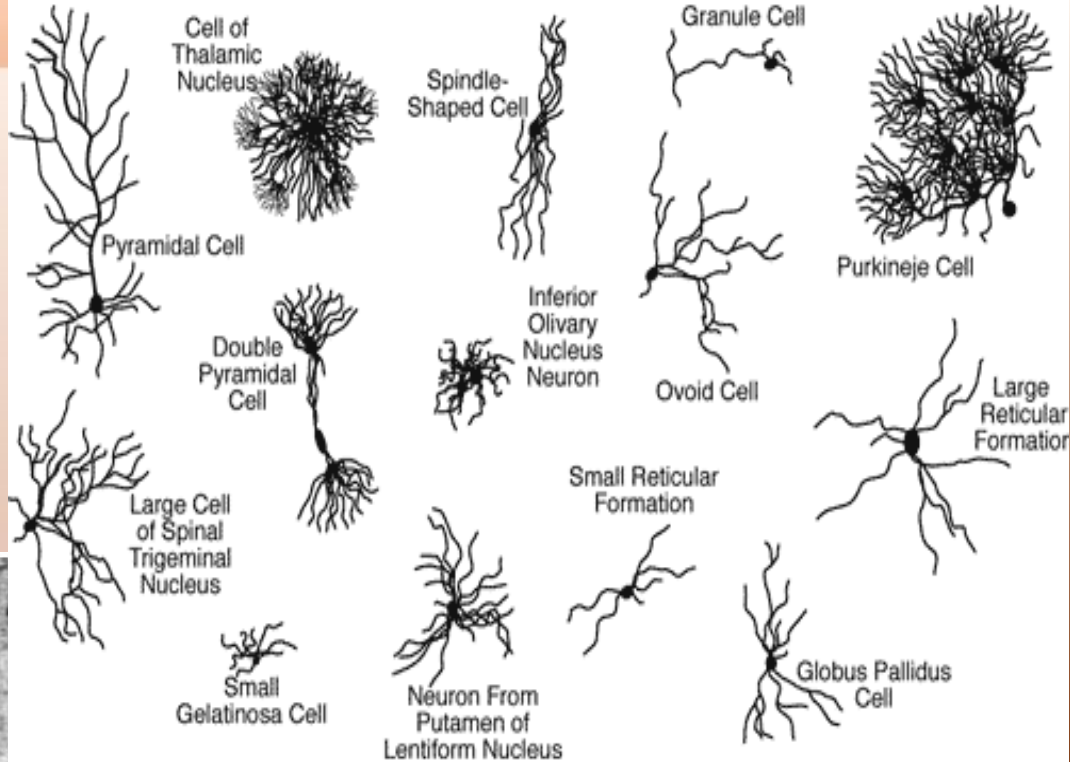
# Types of Neurons

## Basic Neuron Types



Bipolar (Interneuron)    Unipolar (Sensory Neuron)    Multipolar (Motoneuron)    Pyramidal Cell

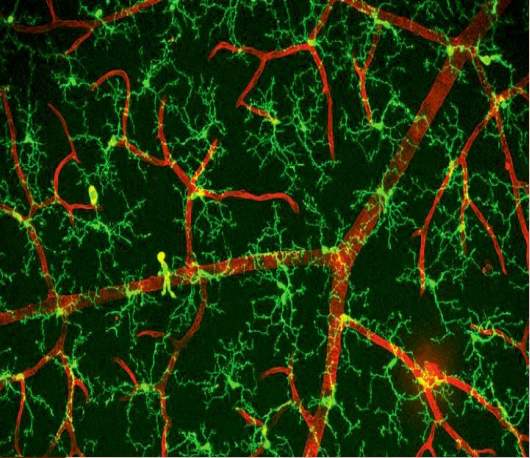
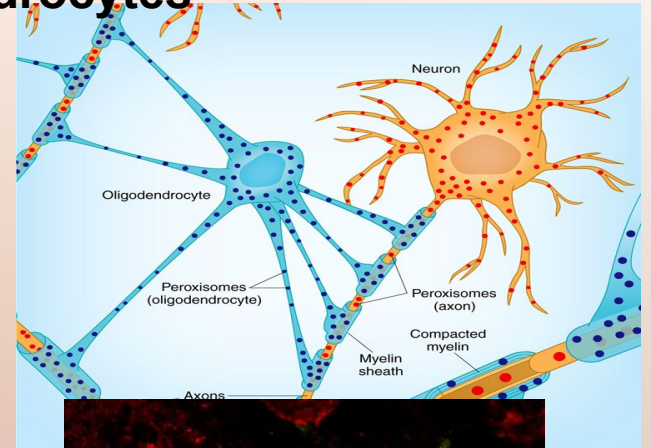
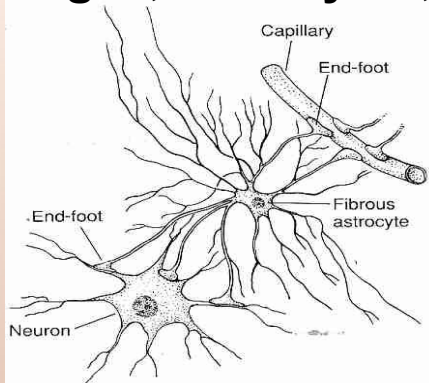
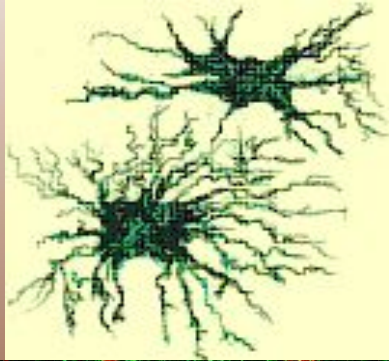
©2001 HowStuffWorks



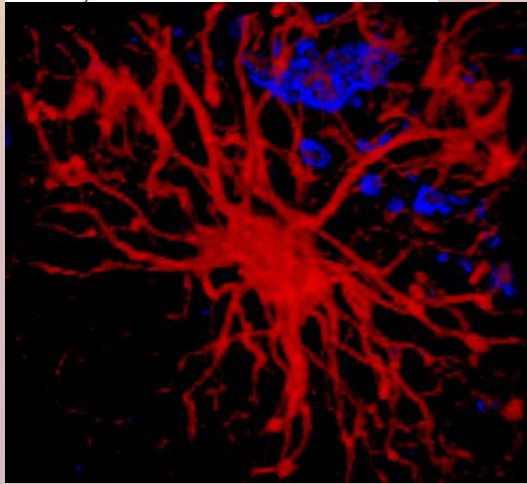


# Other Cell Types in the Brain

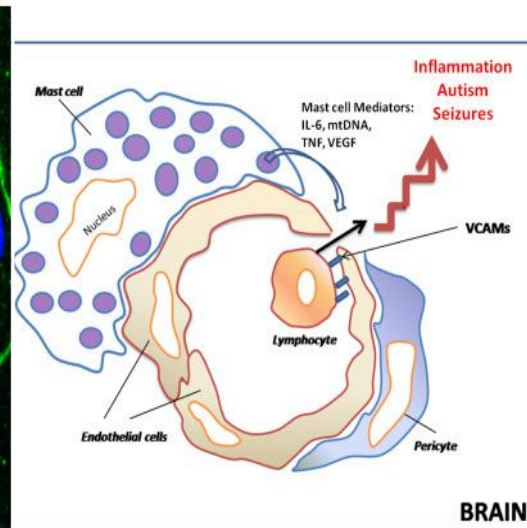
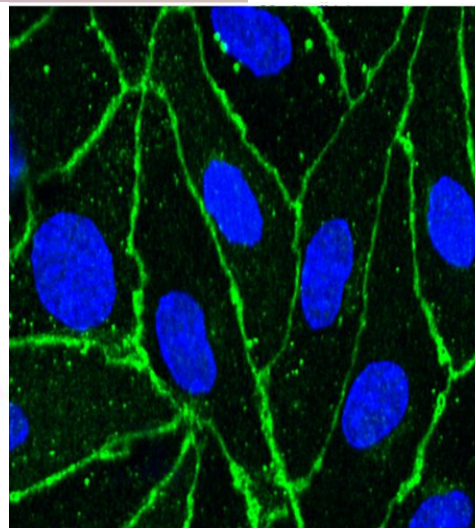
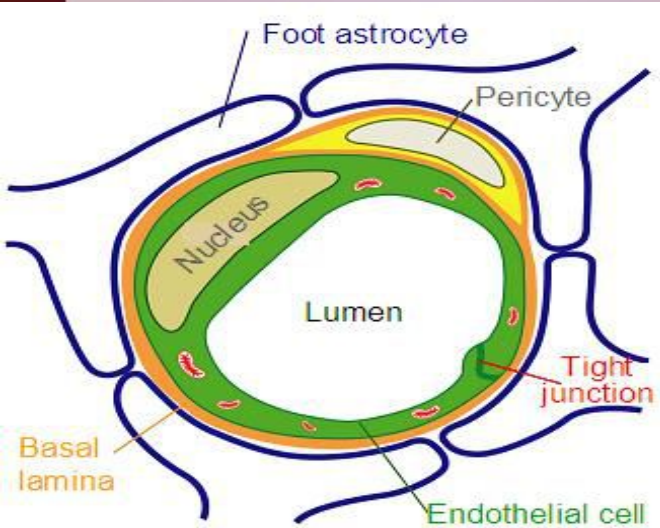
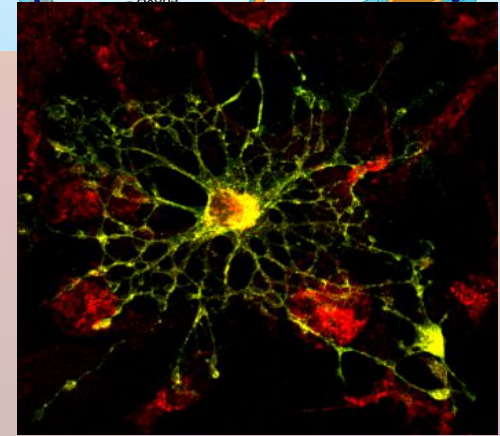
## Glial Cells - microglia, astrocytes, oligodendrocytes



**Endothelial**



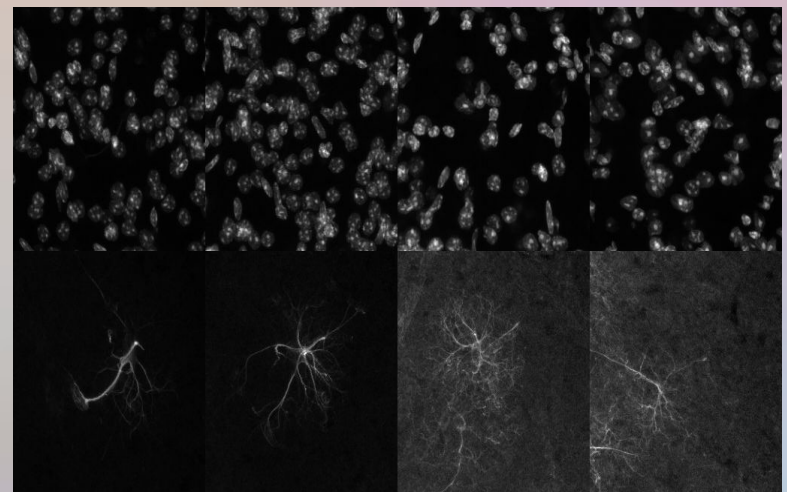
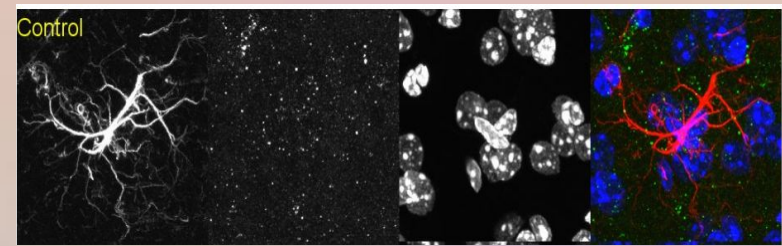
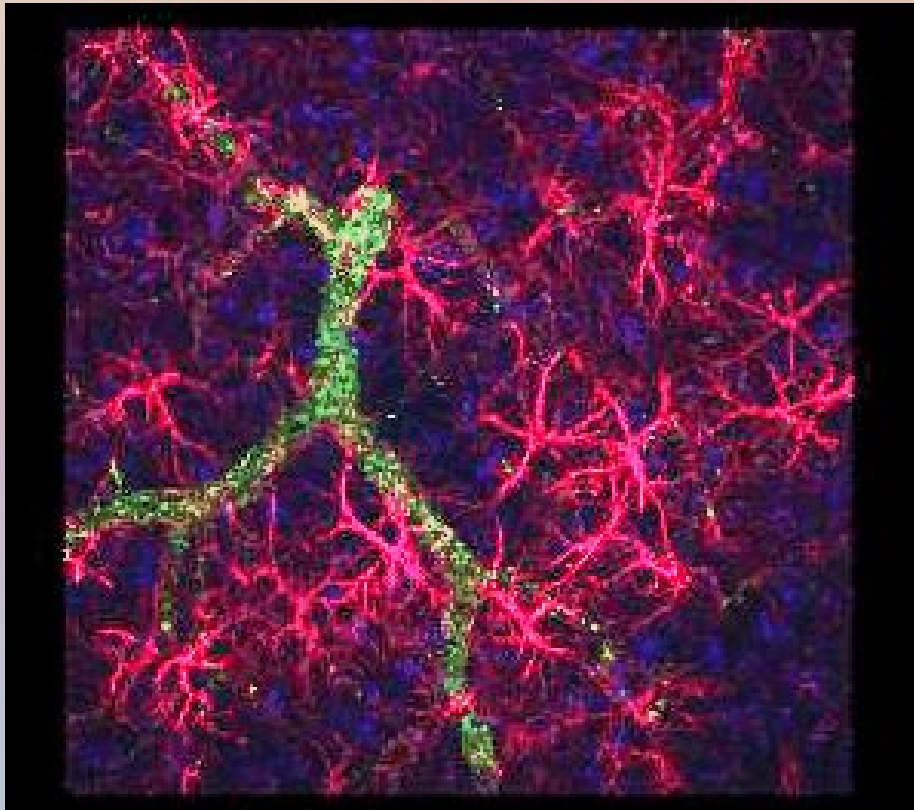
**Mast Cells**



# Imaging Projects

## 2) *Astrocyte analysis*

- automatically segment *Individual* Cells in 3D
- classify cells and provide volume and surface area
- extract and count individual glial cells from larger data
- use and extend existing plugins
- beginning point from existing course

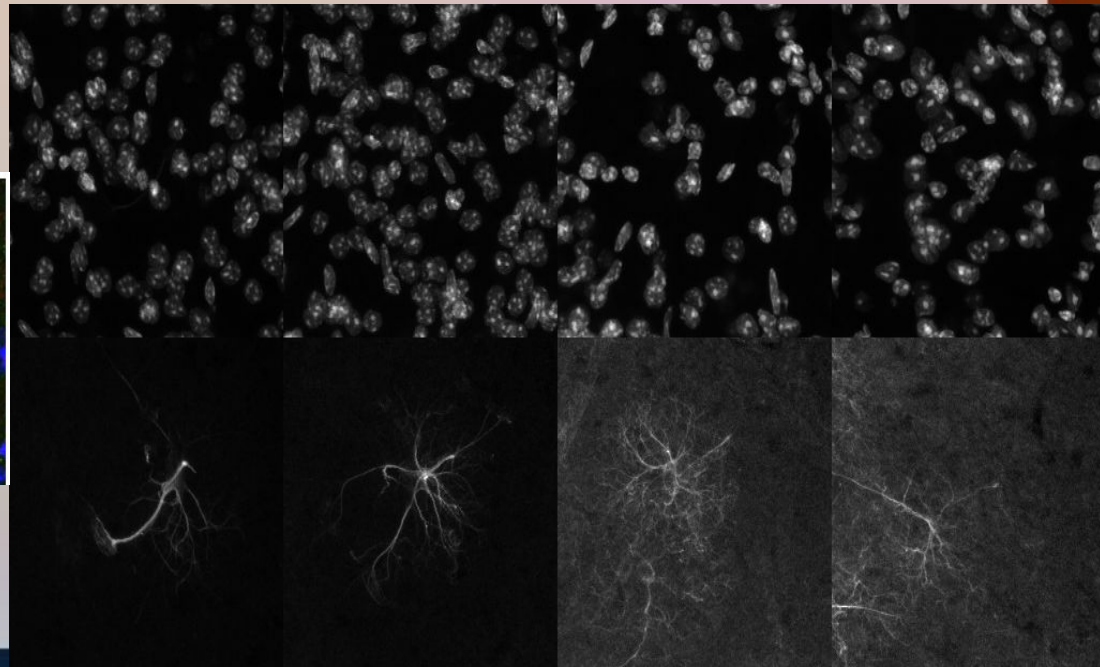
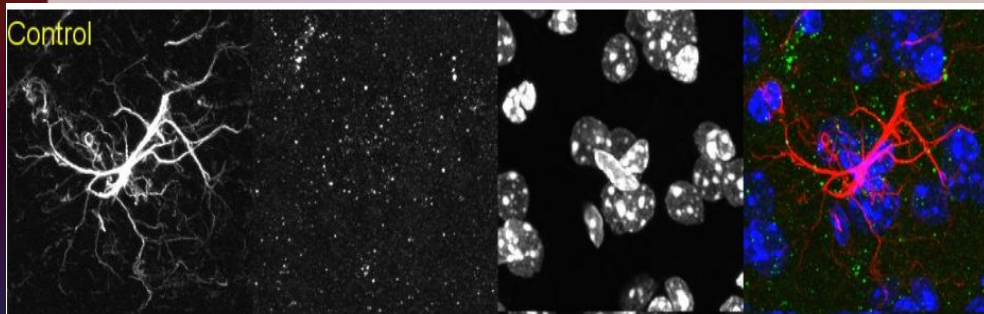




# Imaging Projects

## 2) *Astrocyte analysis*

- automatically segment *Individual Cells* in 3D
- intelligent thresholding → brightness differences?
- segmentation → use of seed points
- extendable to larger data sets with multiple astrocytes
  - using seed points
  - overlapping/touching cell structures?



# Imaging Projects

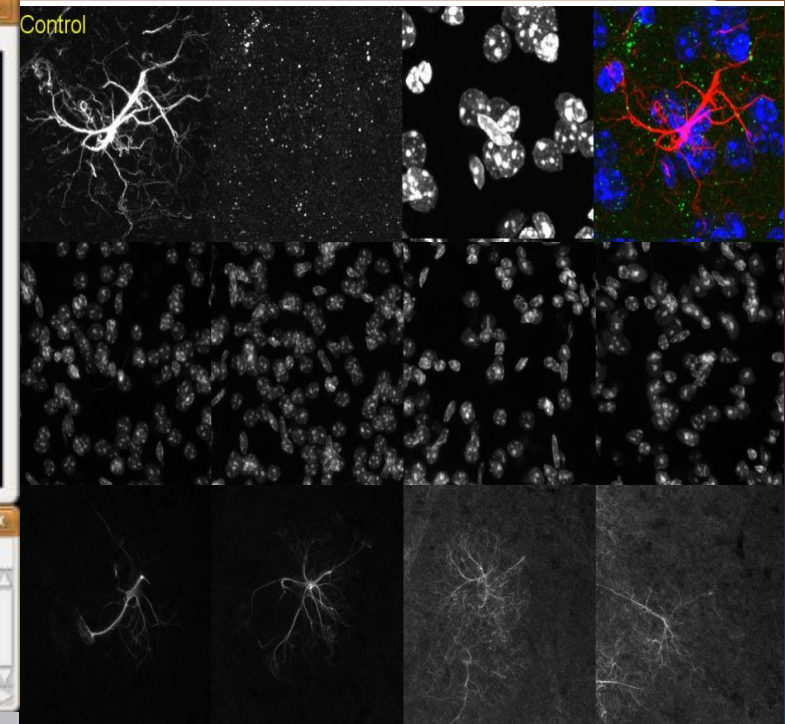
## 2) *Astrocyte analysis*

- classify cells and provide volume and surface area
- existing tools for quantification
- describe number/length of cell extensions
- define cell body
- describe number of bifurcations
- group similar cells for classification
- use existing plugins (---skeletonize3d/analyze skeleton?)



Results

Branch Length	# Triple points	Maximum Branch Length
15	7	72.669
5	2	48.799
7	3	67.113

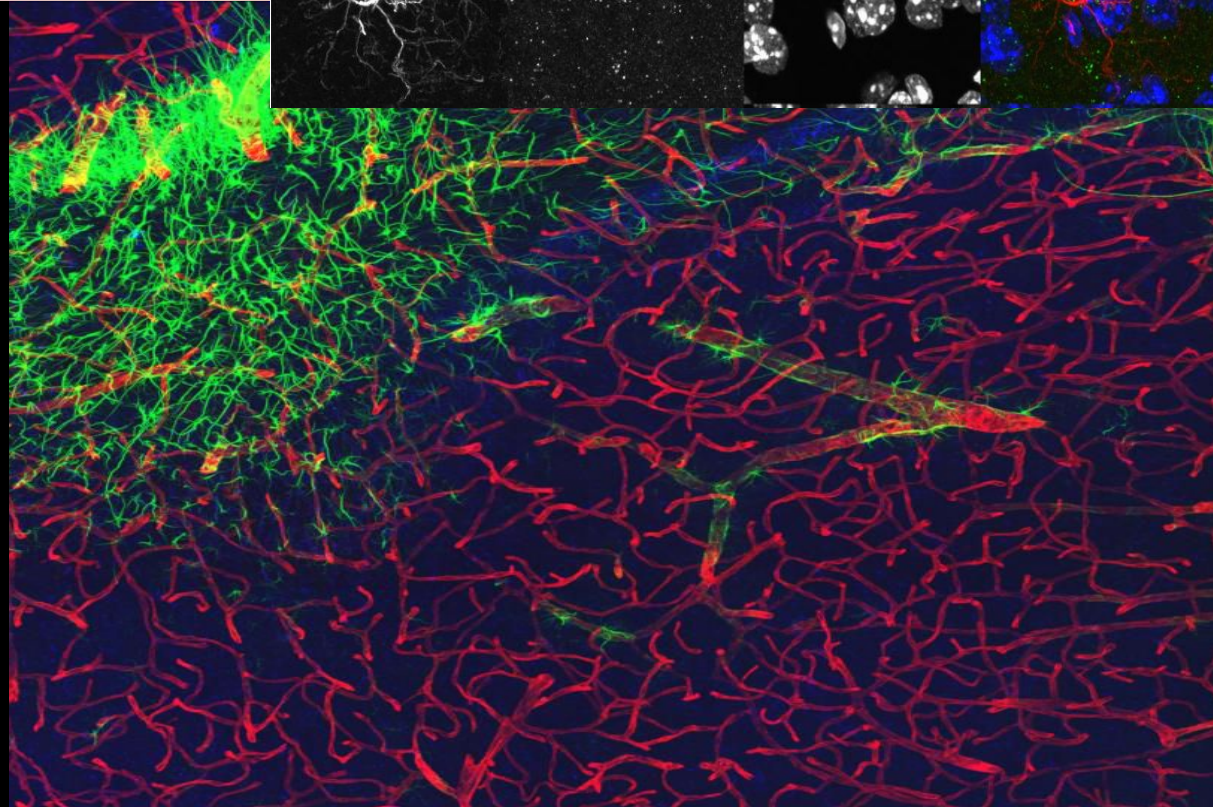
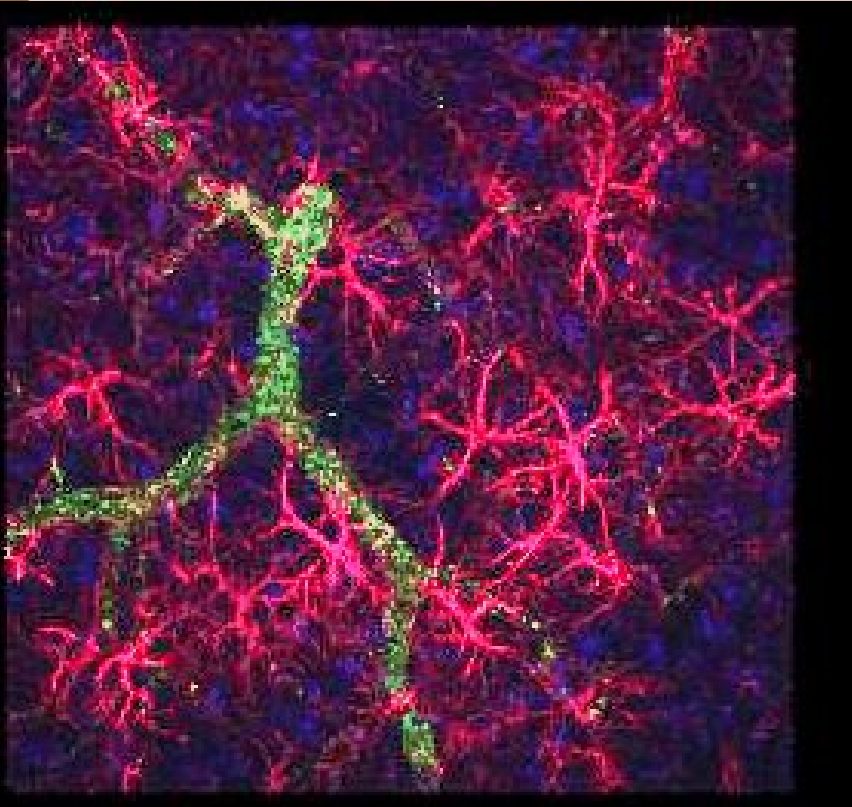
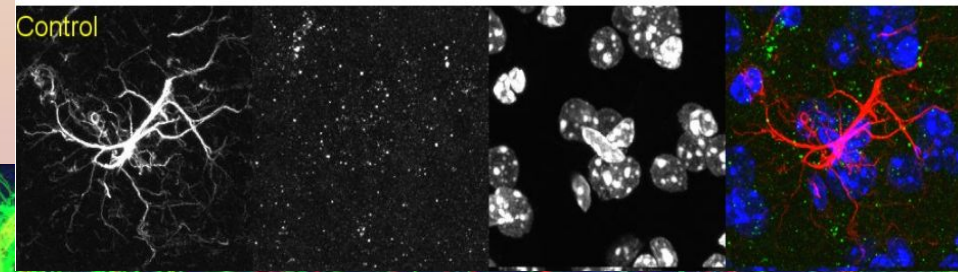




# Imaging Projects

## 2) *Astrocyte analysis*

- extract, classify and count individual glial cells in larger image
- find seed points to start segmentation
  - use other channel information (nuclear stain)
- overlapping glial cells?
  - where to terminate segmentation?

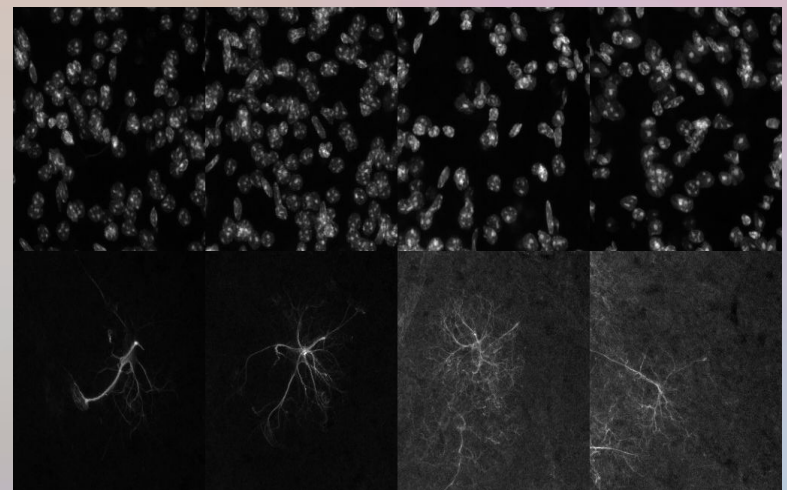
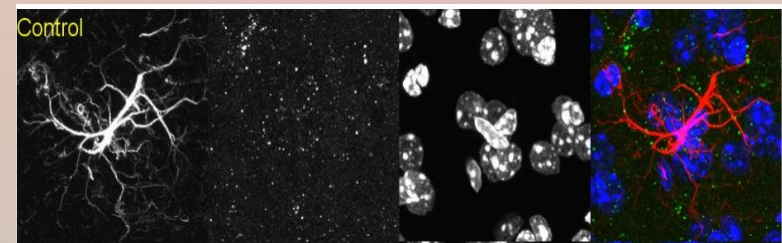
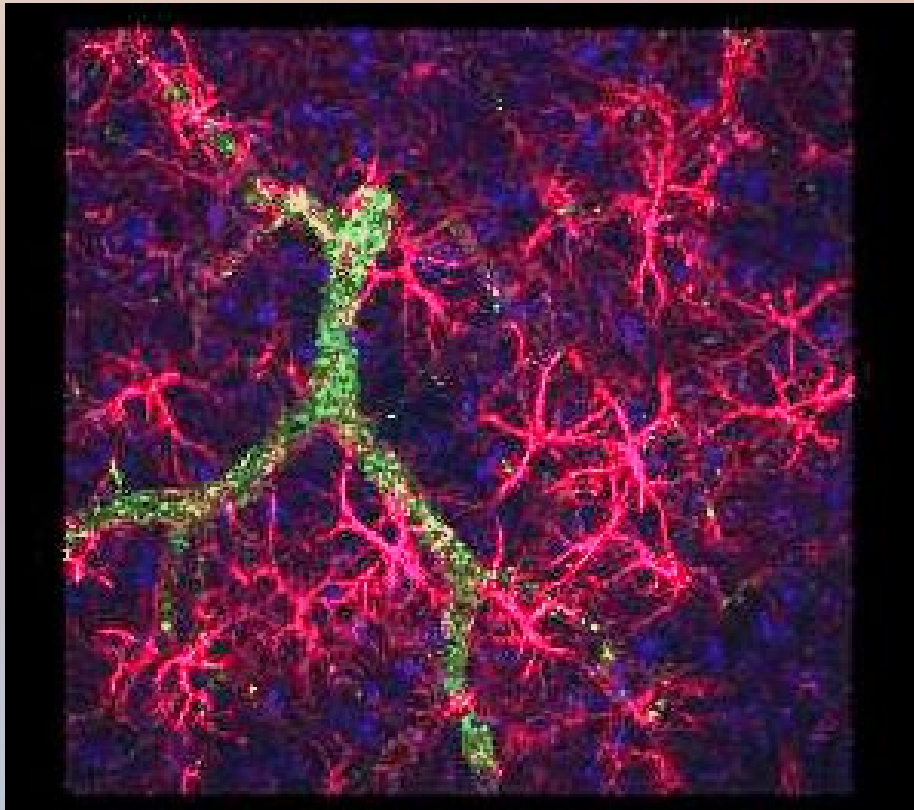




# Imaging Projects

## 2) *Astrocyte analysis*

- automatically segment *Individual* Glial Cells in 3D
- classify astrocytes and provide volume and surface area
- extract and count individual glial cells
- measure degree of overlap with vessels
- use and extend existing plugins



# Existing Tools

- works on single cells
- implementation?
- seed point identification?
- Work on multiple data?



#	# Branches
1	3
2	7
3	5
4	1
5	1
6	3
7	12
8	3

Skeleton ID	Branch length
1	62.897
2	23.778
3	7.196
4	27.388
5	20.242
6	19.510
7	11.217
8	6.464
9	2.414
10	1.000
11	41.245
12	19.510
13	5.146
14	3.464
15	2.000
16	32.438
17	33.459
18	28.534
19	18.535
20	6.975
21	21.828
22	15.535
23	13.899
24	12.474
25	8.975
26	8.742
27	7.196
28	5.243
29	4.560
30	4.560
31	3.732
32	1.000
33	11.071
34	4.560
35	2.732

# Imaging Projects – *Plugins Creation*

## MICROSCOPY

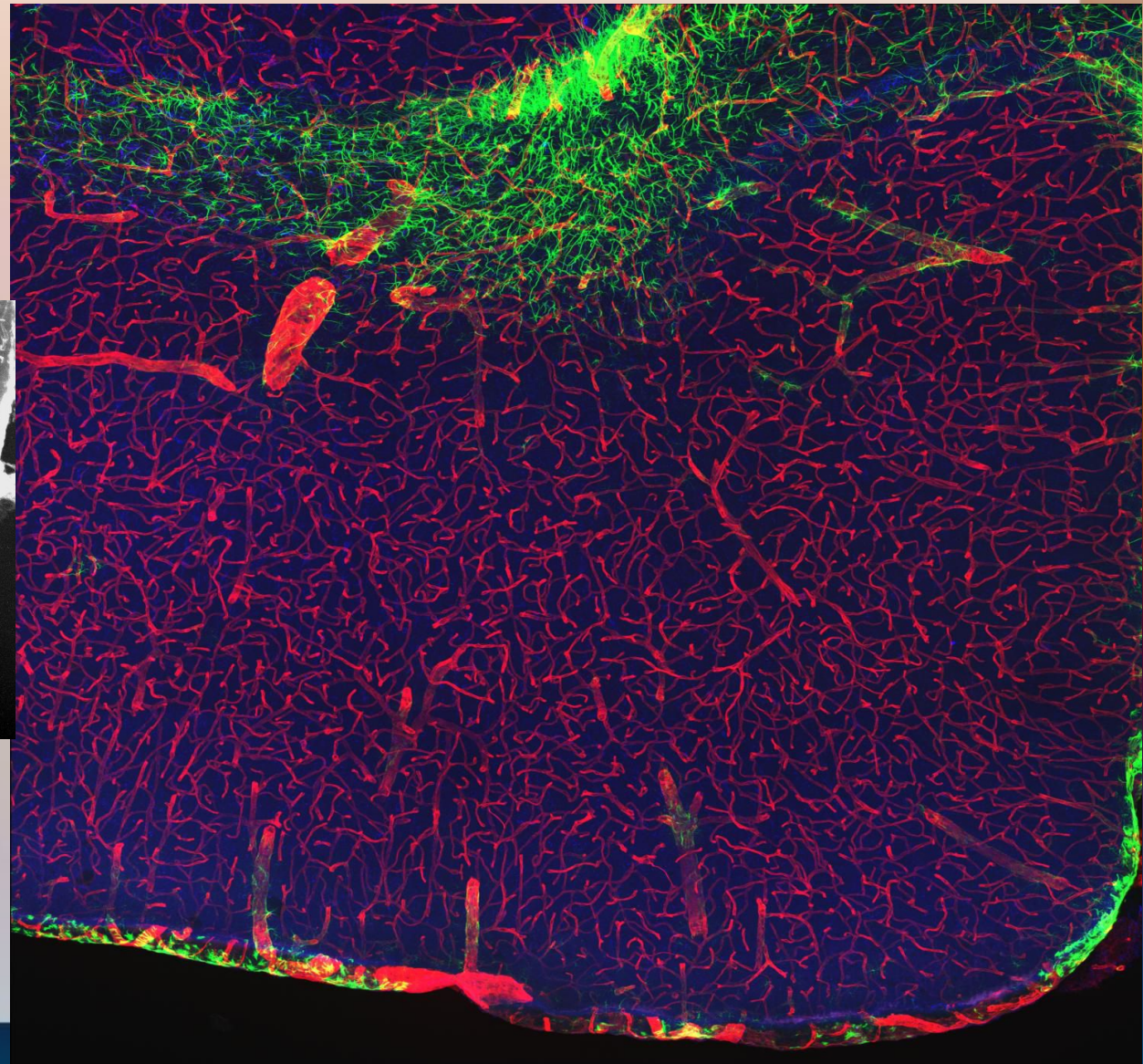
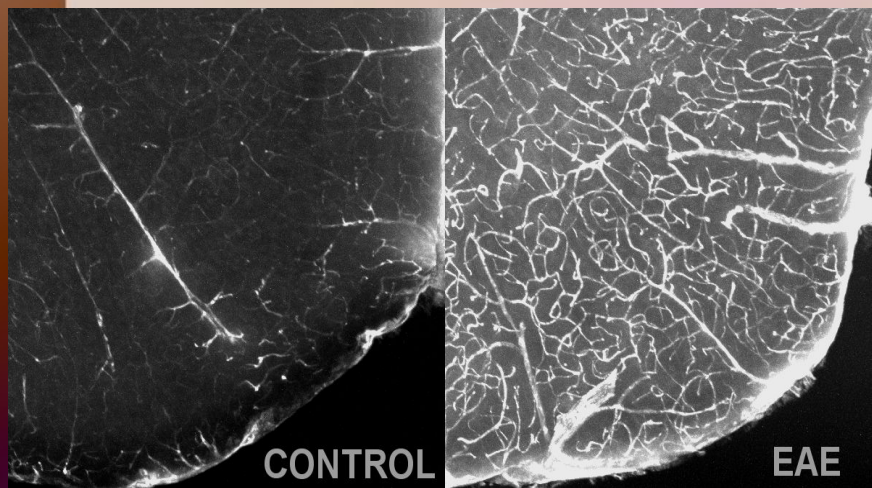
- 1) Tiled data reconstruction and correction
- 2) Astrocyte Cell classification
- 3) *Microvessel analysis*



# Imaging Projects

## 3) Microvessel analysis

- segment microvessels in 3D
- classify vessel segments
- surface area/volume
- segment width, density
- identify branching patterns



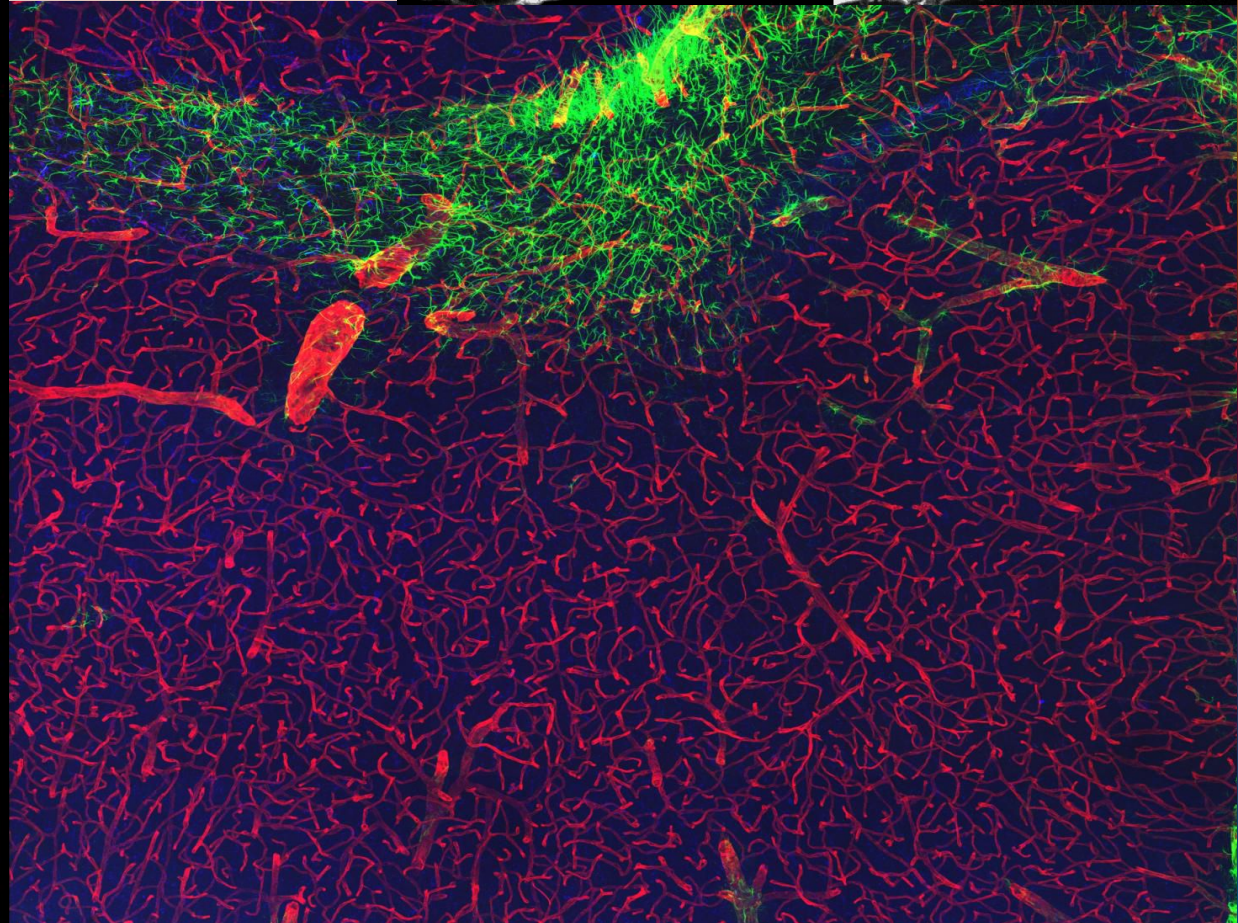
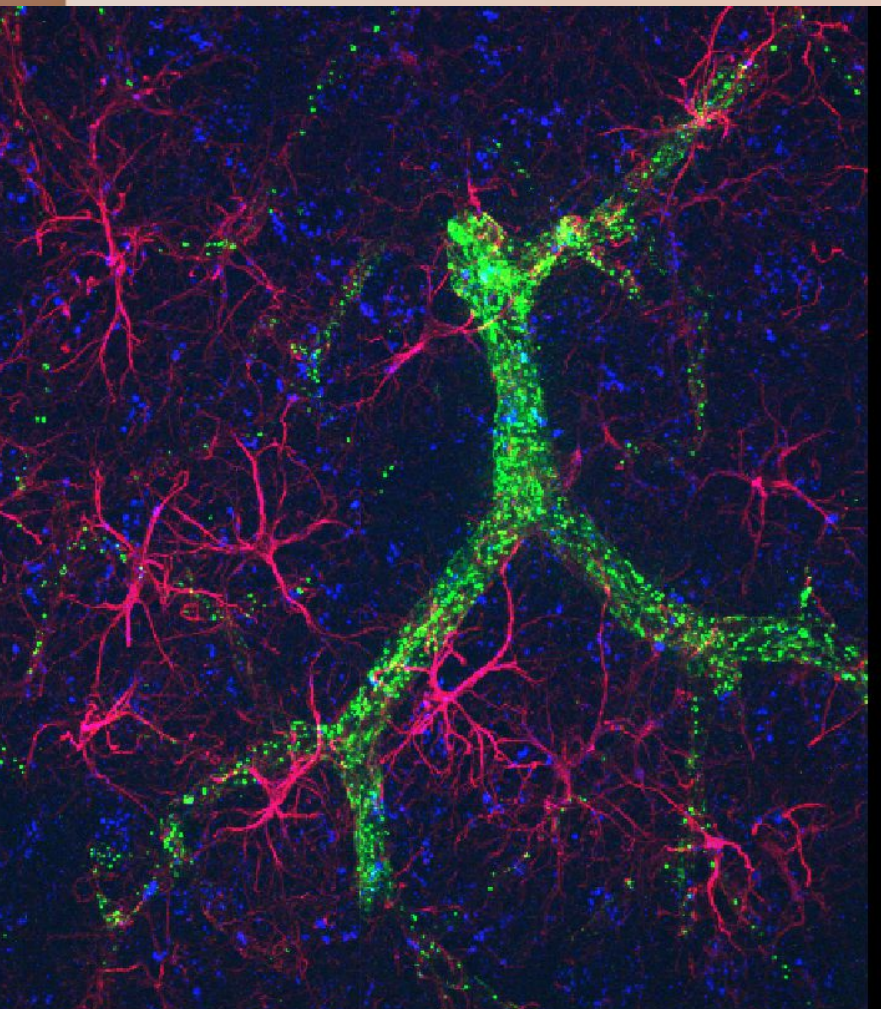
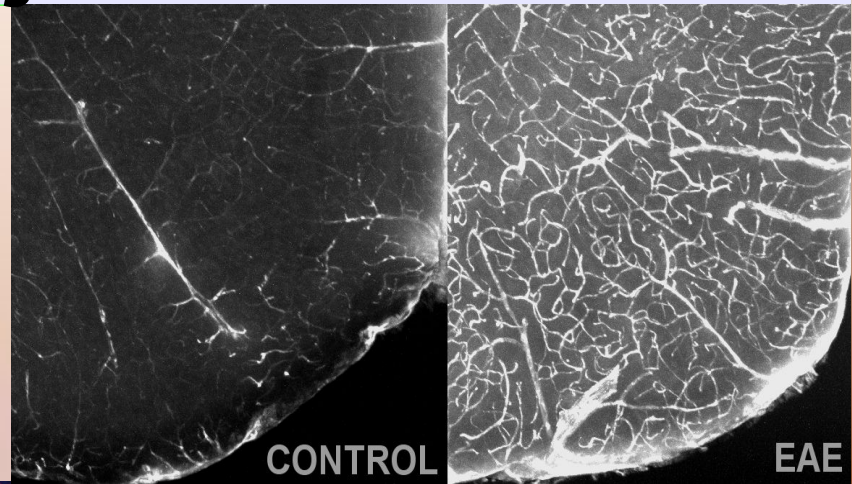


# Imaging Projects

## 3) Microvessel analysis

Segment microvessels in 3D

- intelligent thresholding
- resistant to partial staining techniques
- transected segments?



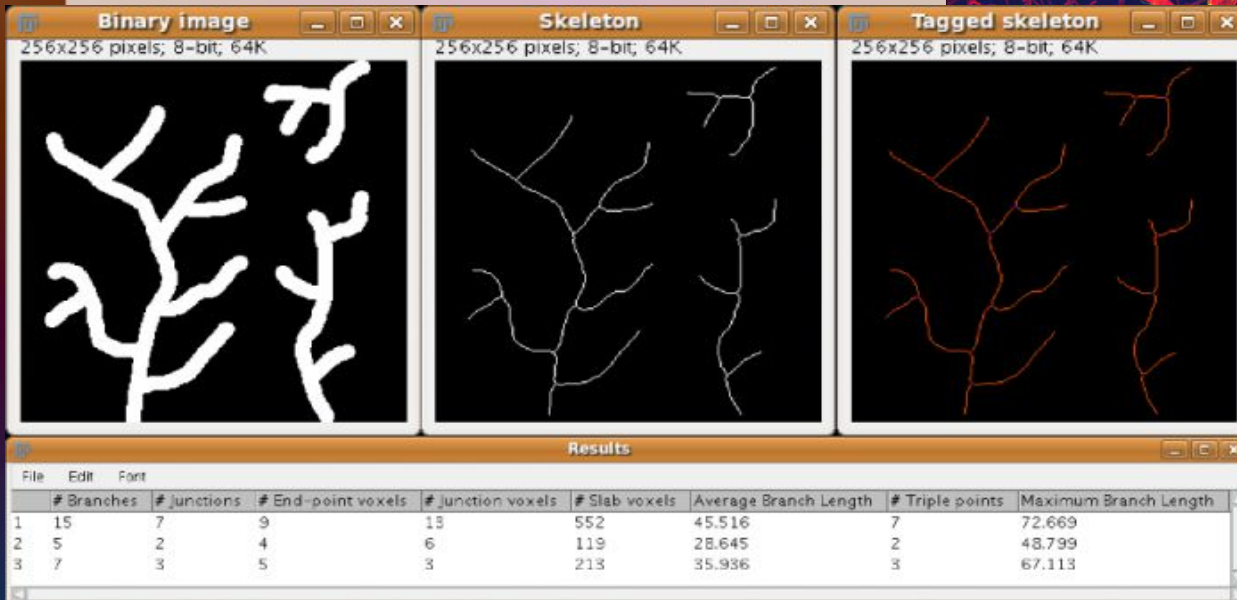
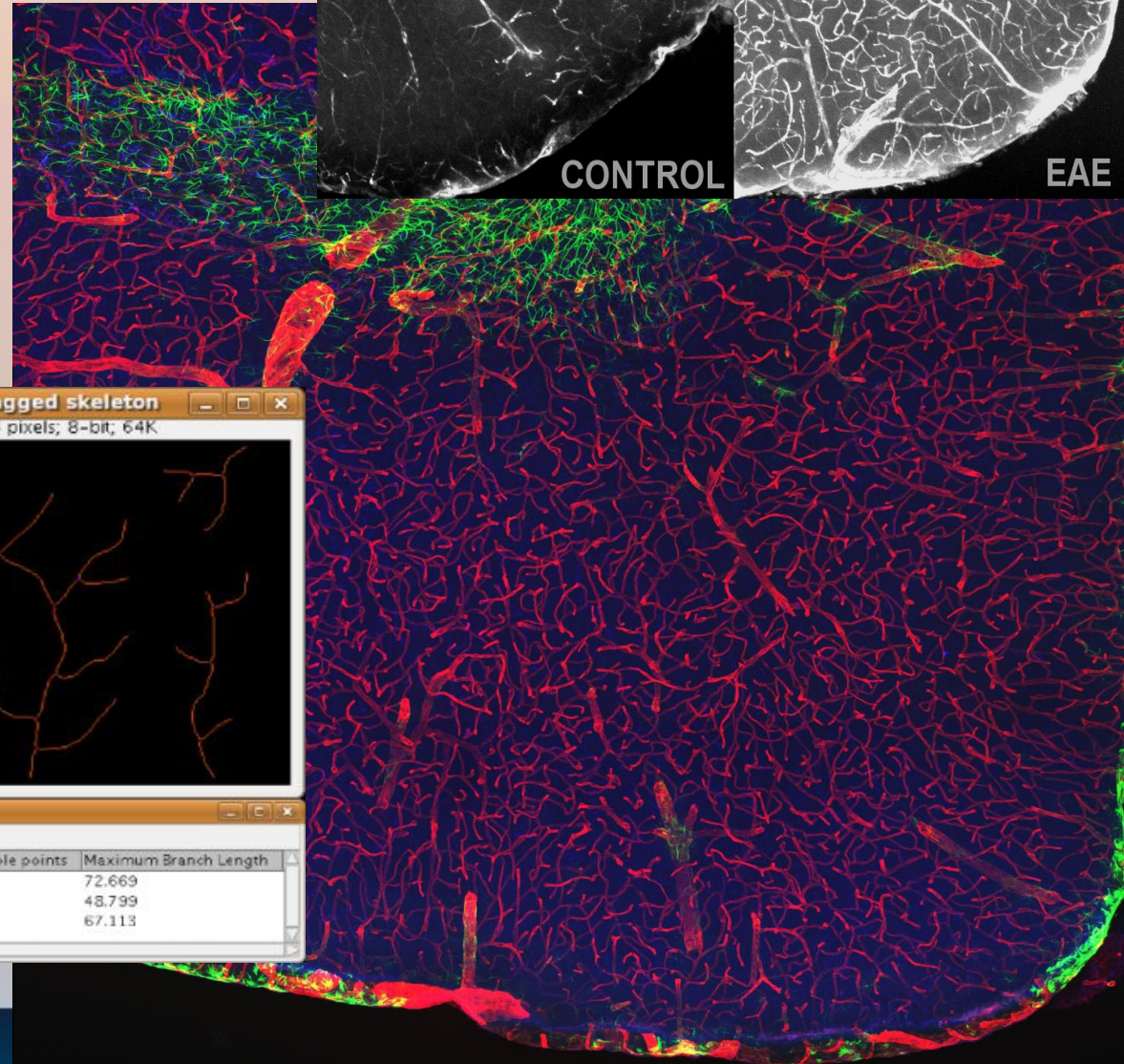
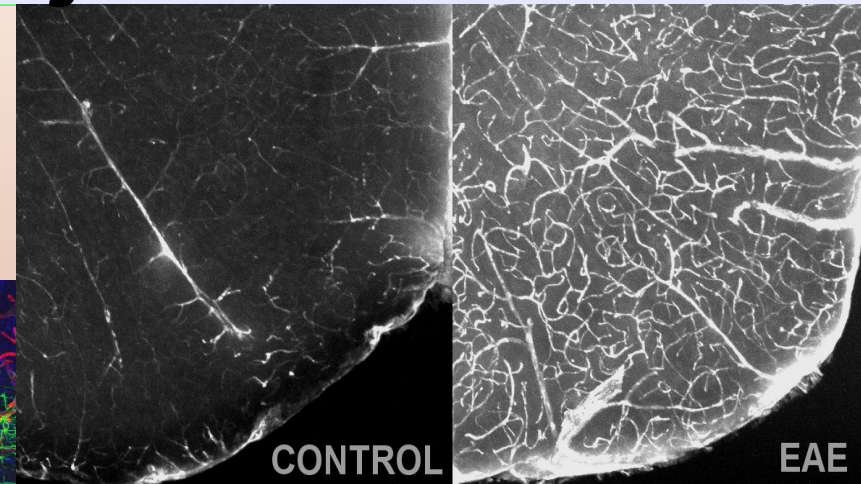


# Imaging Projects

## 3) Microvessel analysis

### Classify vessel segments in 3D

- number of segments
- surface area/volume
  - existing tools
- segment width
- vessel density
- identify branching patterns
  - existing plugins



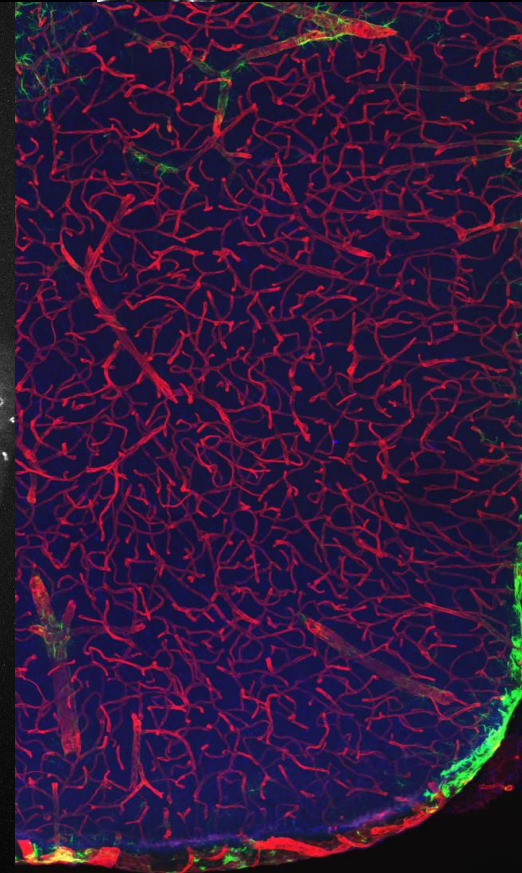
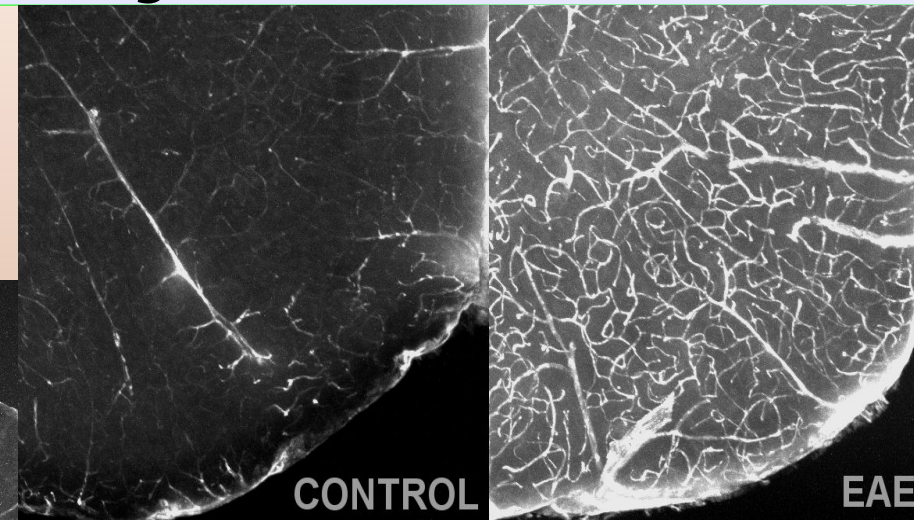
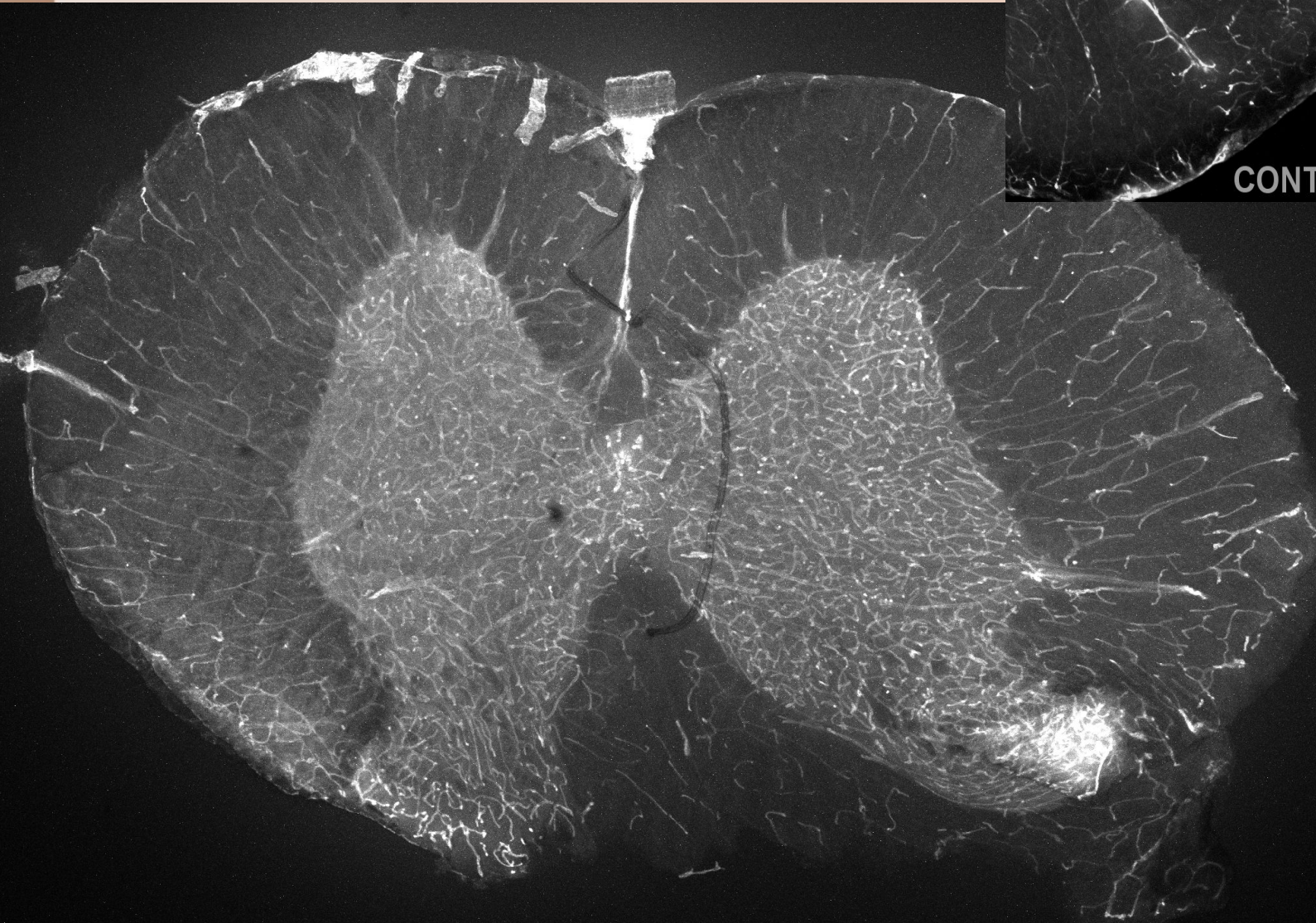


# Imaging Projects

## 3) Microvessel analysis

Test on Multiple datasets

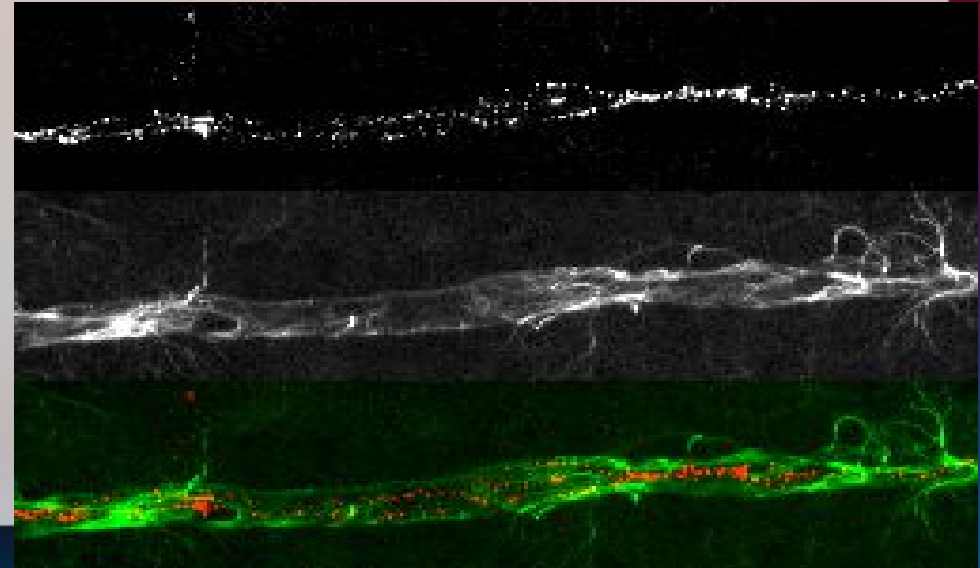
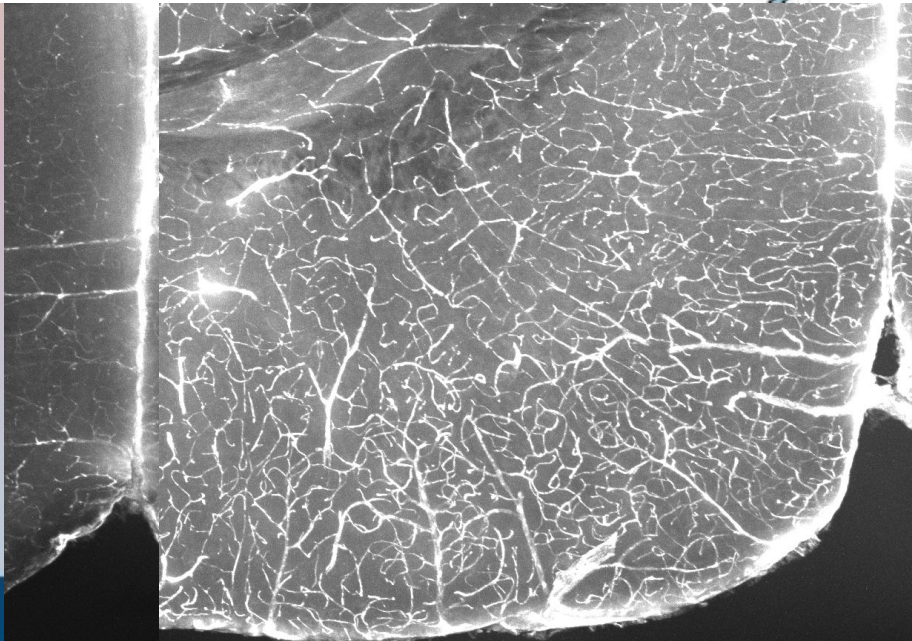
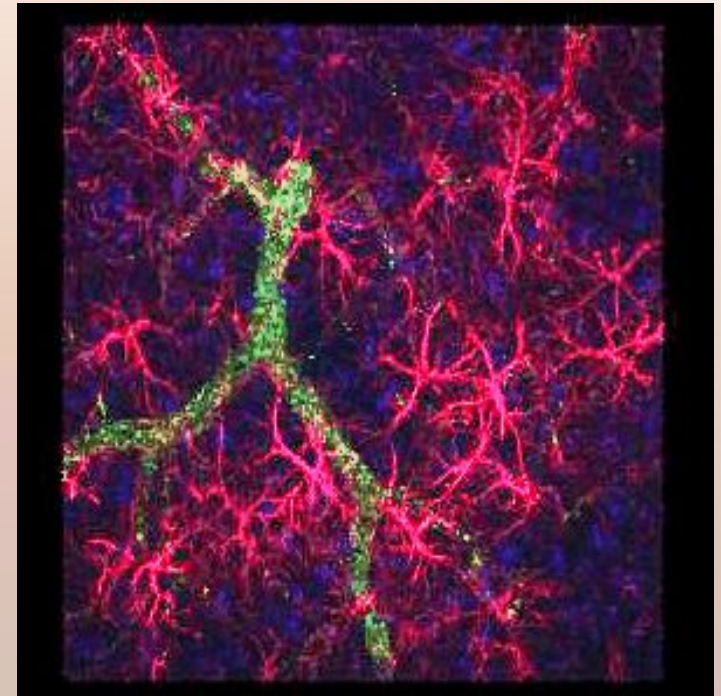
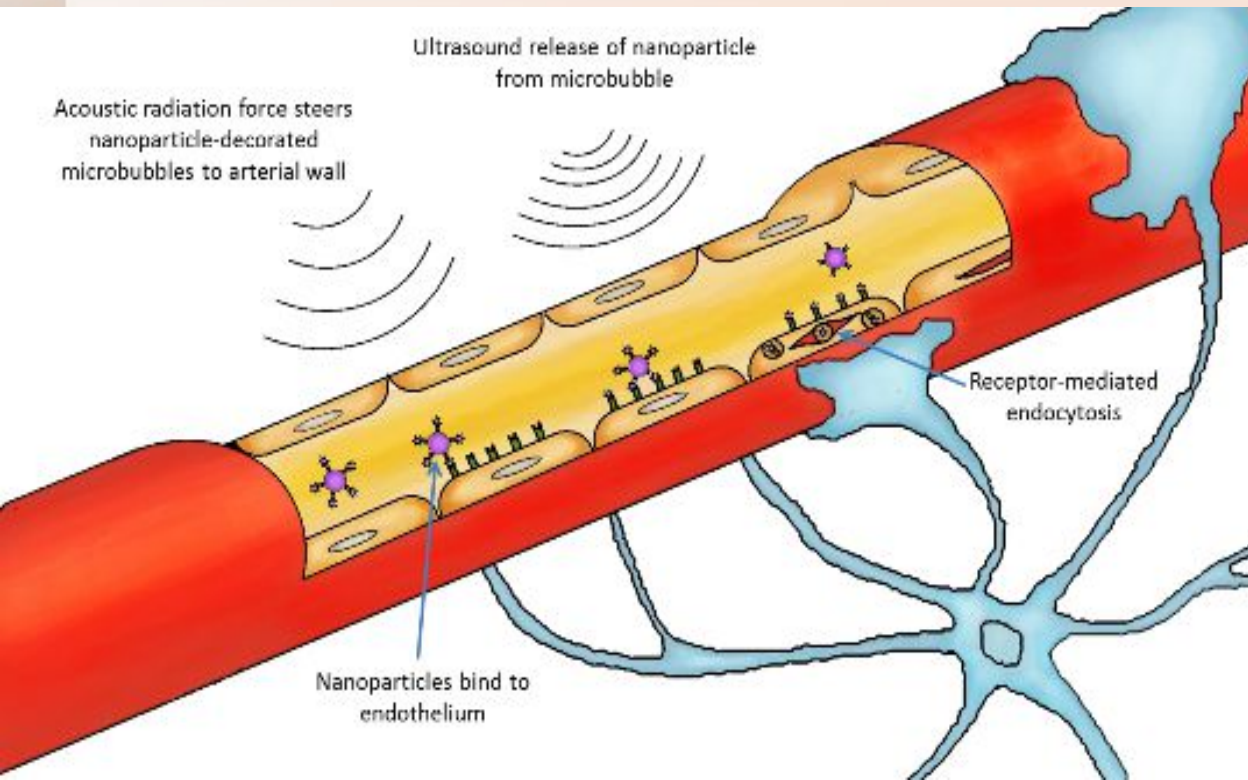
Provide relevant metric output





# Functional Units in the Brain

## Blood Brain Barrier

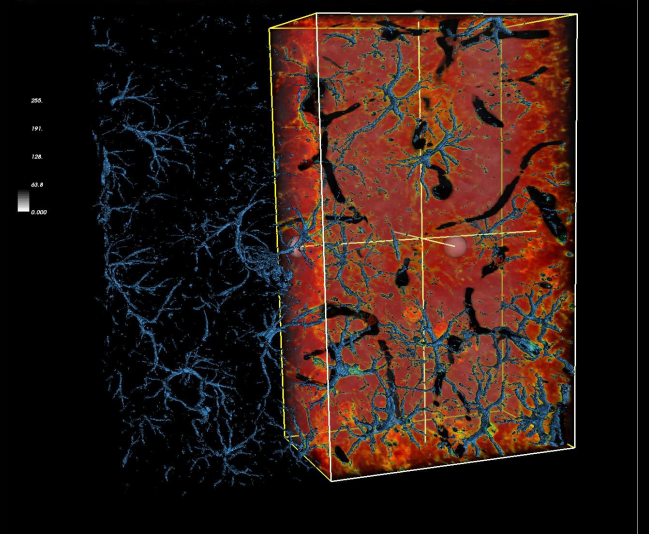
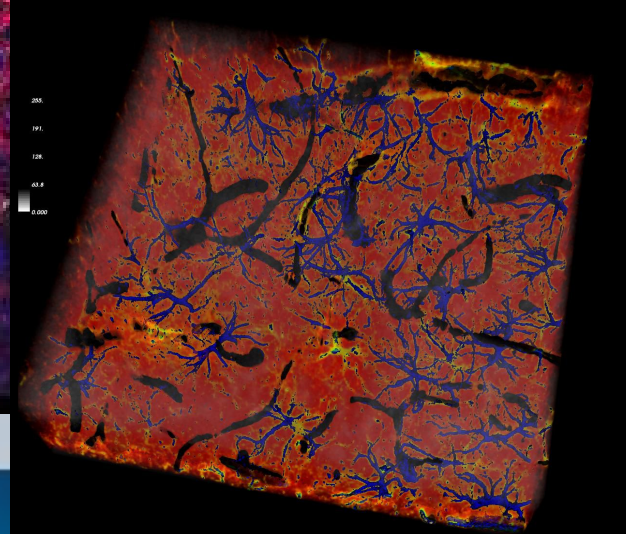
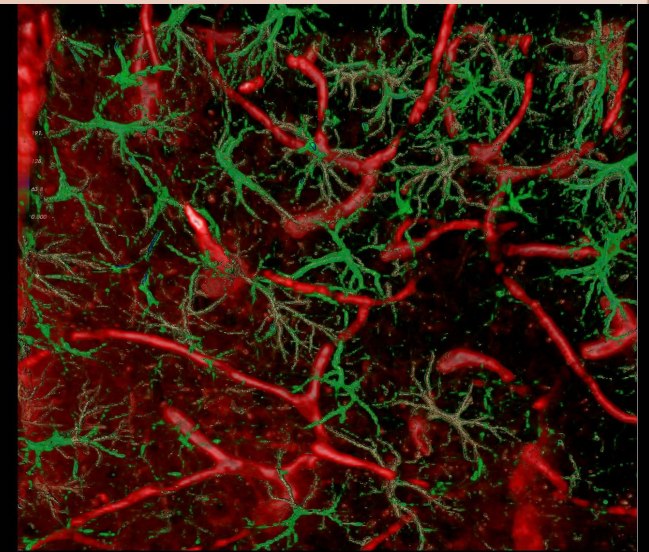
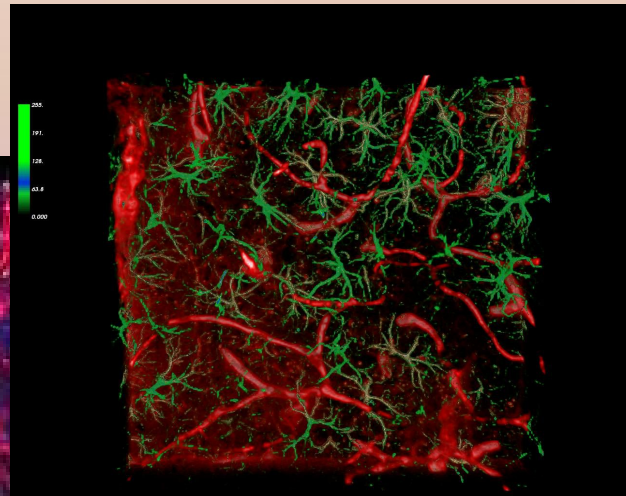
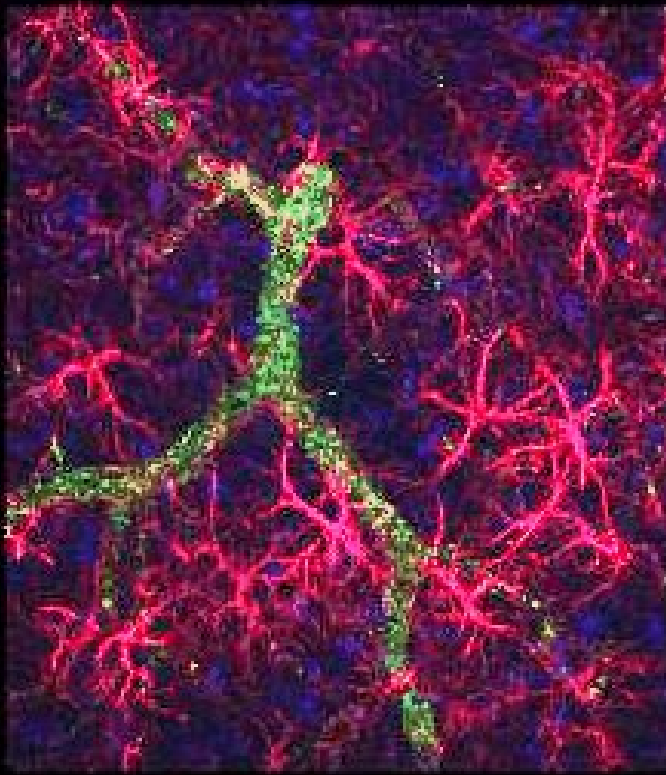




# Imaging Projects

## 3) Microvessel *analysis*

-measure degree of astrocyte overlap with vessels in secondary channel



# Existing Tools

- vessels of multiple sizes?
- degree of overlap with additional channel?  
(mask creation)

Tube radius (micrometer)	6	1) The first two parameters should both be set to the approximate expected radius of the blood vessels. They control the closing of the tube (tube filling) and the filament pre-filter. 2) Vessel threshold: Sets the intensity sensitivity of the detection (the lower the more sensitive). 3) "Remove branched with end-points": Skeleton branches leading to an end-point can optionally be removed. 4) "Dilate Skeleton for viewing by (pixels)": Only used for visualization in the 3D viewer. 5) "Number of threads": Many filters are multi-threaded, for best performance adjust this value to the number of cores of your CPU(s) or twice this number if your architecture supports hyperthreading. 6) "Save intermediate steps": Save all the intermediate results (stacks) to disk.
Vessel radius (micrometer)	6	
Vessel threshold	8	
Minimal Vessel Volume (pixels)	1000	
<input type="checkbox"/> Remove branches with end-points		
Dilate Skeleton for viewing by (pixels)	2	
Number of threads	8	
<input type="checkbox"/> Save intermediate steps?		
<input type="button" value="OK"/> <input type="button" value="Cancel"/>		

**Results**

The results consist of:

- 1) A label mask of connected components (Fig 4.2).
- 2) A labelled skeleton stack showing the skeleton, the branching and end-points (Fig 4.3).
- 3) A log holding some statistics on the blood vessel network (Fig 4.4) such as the average blood vessel cross-section area and diameter, the total length of the network, the number of branches and the overall vascularization density (volume fraction occupied by the blood vessels).

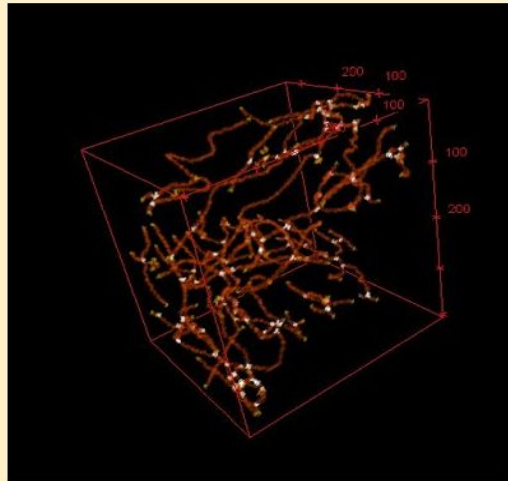


Fig 4.2 Skeleton with end-points (white) and branching points (yellow)

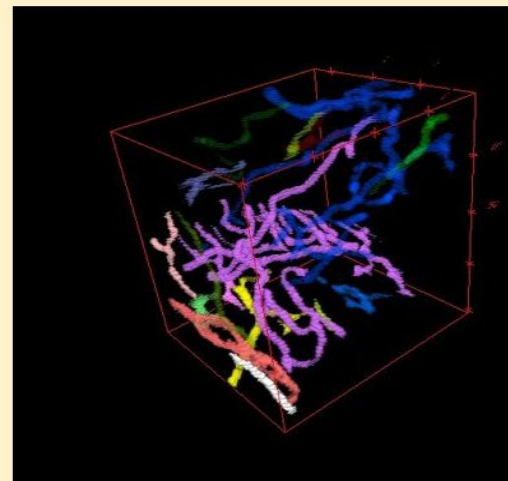
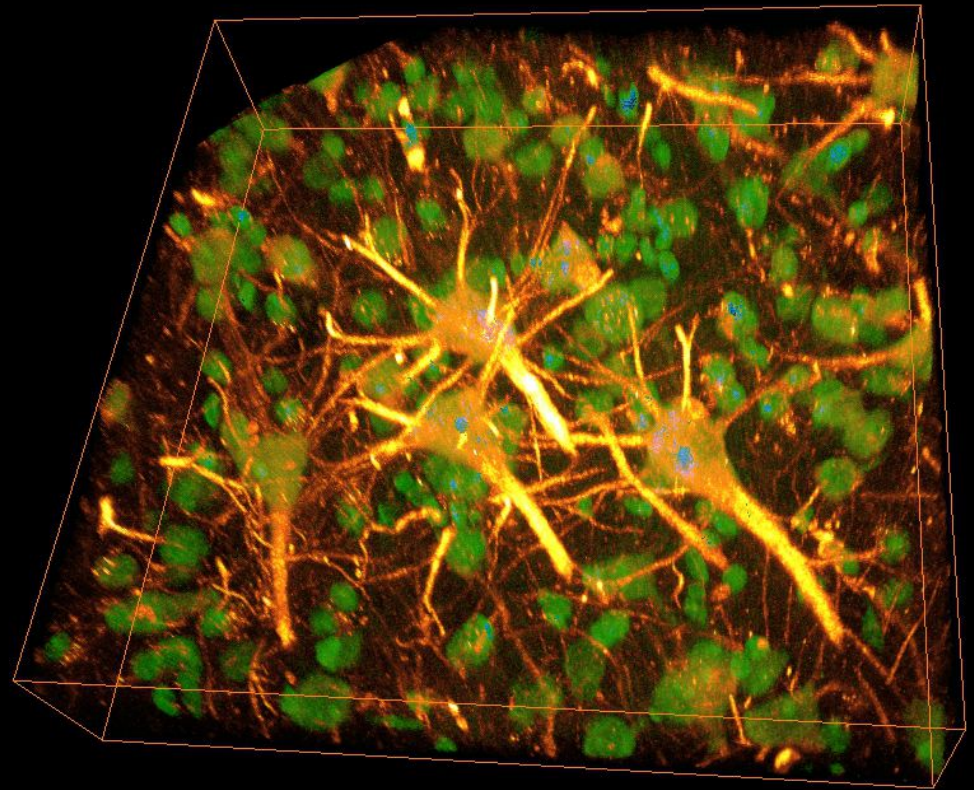
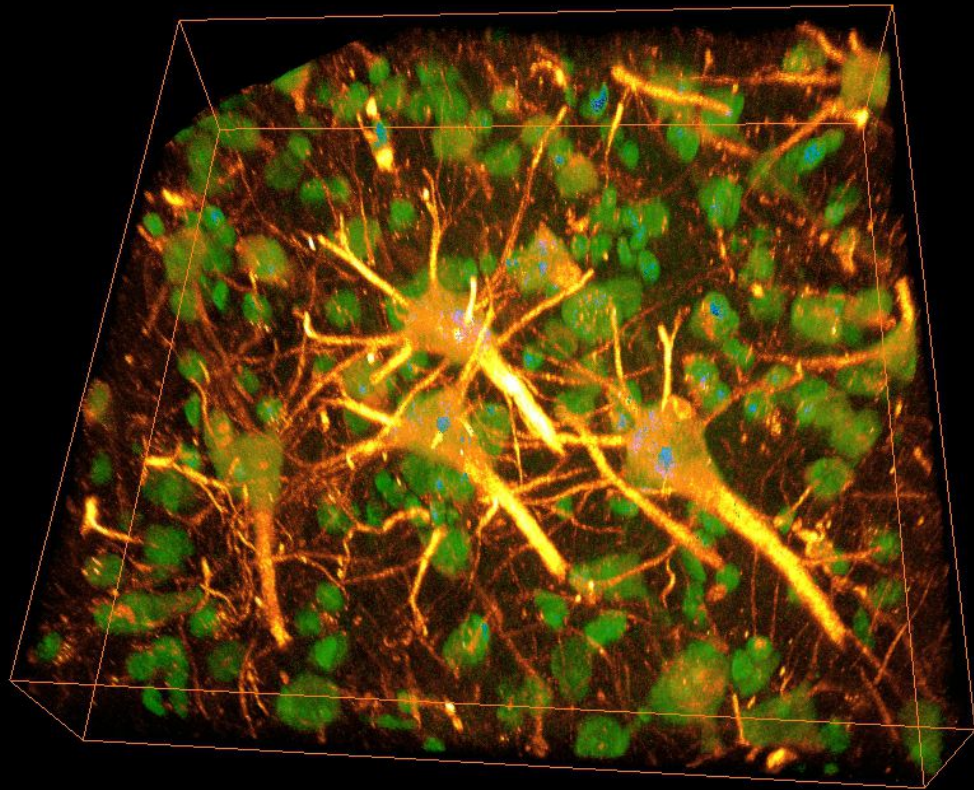


Fig 4.3 Connected components label mask







# Existing Tools

## DATABASES

[Cell Centered Database \(CCDB\)](#),

[SenseLab Project](#)

[L-Neuron](#)

[Neuromorph](#)

[brainmaps.org](#)

[Neuroscience Information Framework \(NIF\)](#)

[Connectome Project](#) .

## Free and Commercial Software

[cellprofiler](#)

[Xvoxtrace](#)

[Neurolucida](#)

[Imaris](#)

[Amira](#)

[SSECRETT and NeuroTrace](#)

## ImageJ/FIJI

[NeuronJ](#) , [Simple Neurite Tracer](#)

[Find Connected Regions](#)

[Neuron Morphology](#) , [TrakEM2](#)

[Simple Interactive Object Extraction \(SIOX\)](#)

[Advanced Weka Segmentation](#)

[Trainable Segmentation plugin](#) .

# Existing Tools

## DATABASES

[Cell Centered Database \(CCDB\)](#),

[SenseLab Project](#)

[L-Neuron](#)

[Neuromorph](#)

[brainmaps.org](#)

[Neuroscience Information Framework \(NIF\)](#)

[Connectome Project](#) .

## ImageJ/FIJI

[NeuronJ](#) , [Simple Neurite Tracer](#)

[Find Connected Regions](#)

[Neuron Morphology](#) , [TrakEM2](#)

[Simple Interactive Object Extraction \(SIOX\)](#)

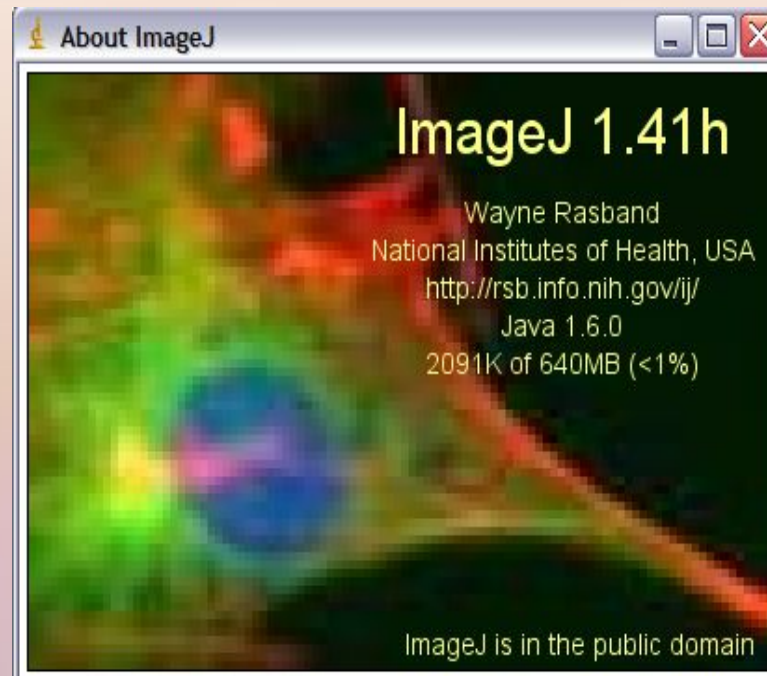
[Advanced Weka Segmentation](#)

[Trainable Segmentation plugin](#) .



# IMAGEJ

<http://rsbweb.nih.gov/ij/>



## Plugins and Macros

<http://rsbweb.nih.gov/ij/plugins/index.html>

<http://rsbweb.nih.gov/ij/developer/macro/macros.html>

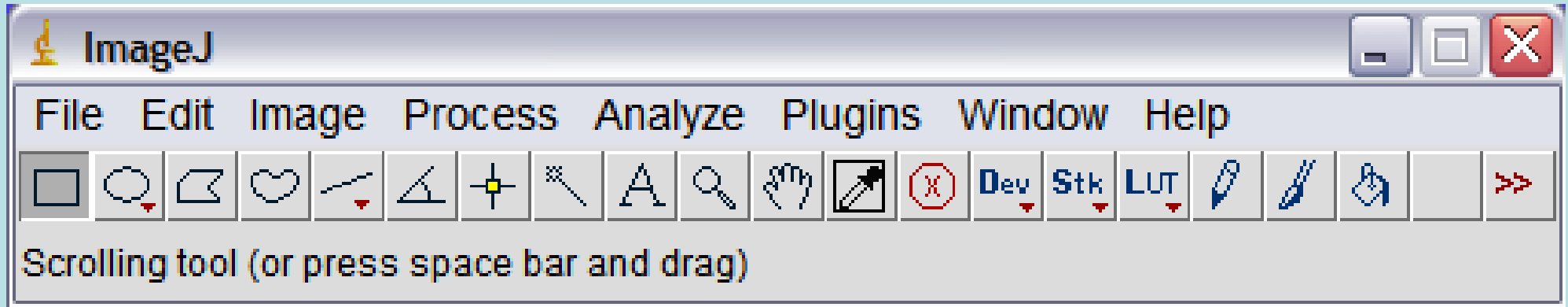
## Examples

<http://rsbweb.nih.gov/ij/docs/examples/index.html>

## Forum

<http://imagej.1557.n6.nabble.com/>

# Introduction to the Main Menu

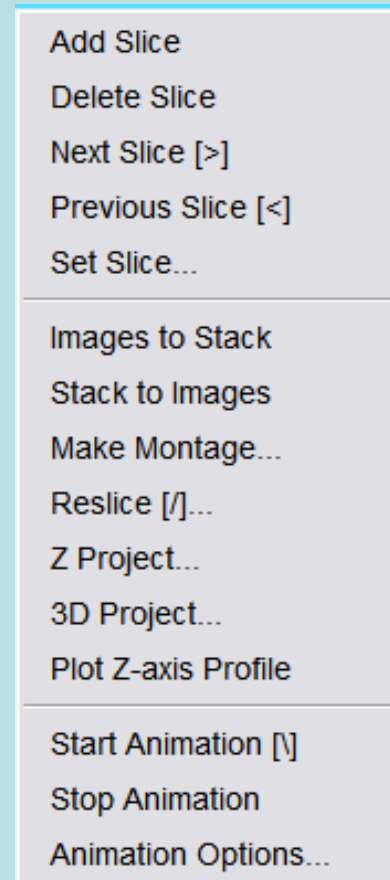
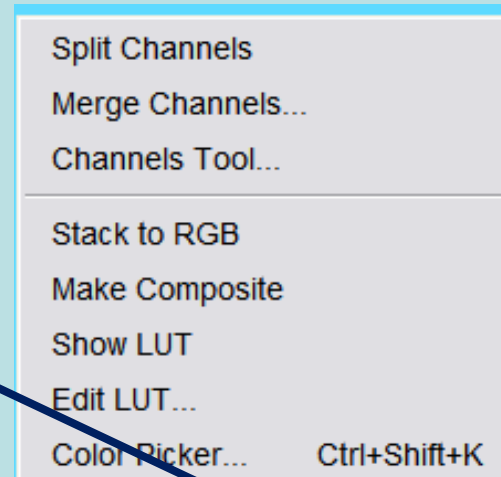
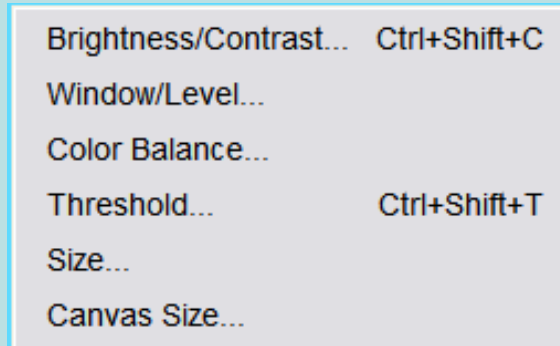
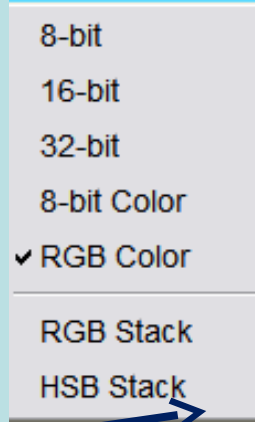
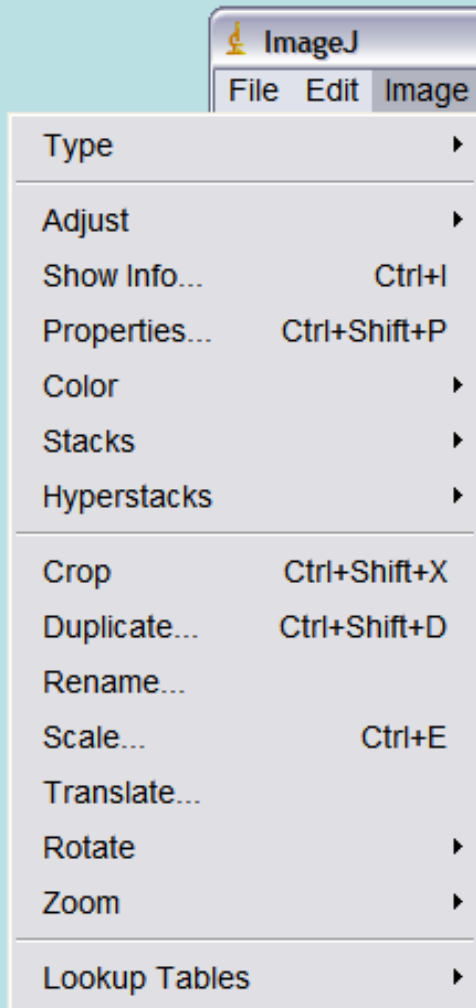


Of these, we'll concentrate on:

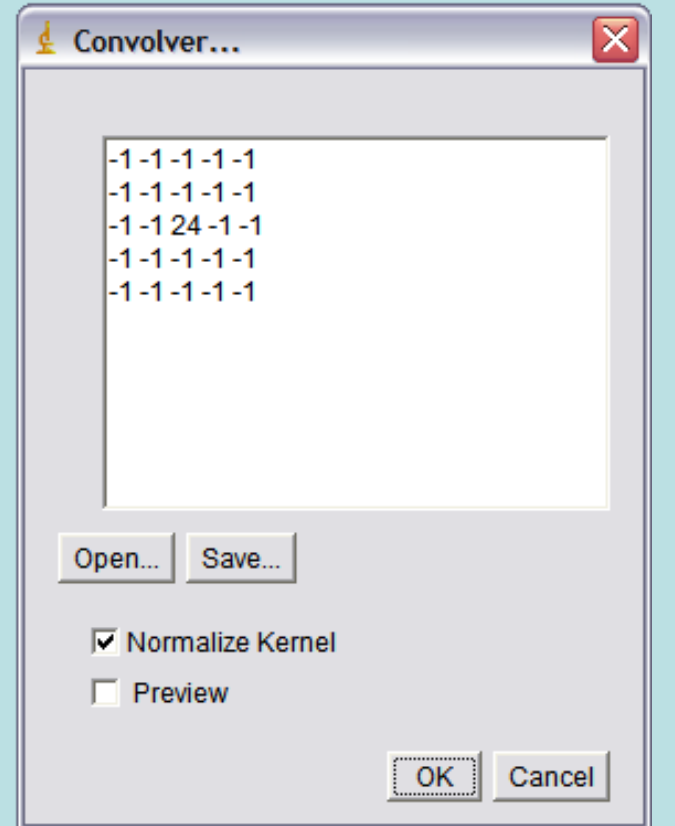
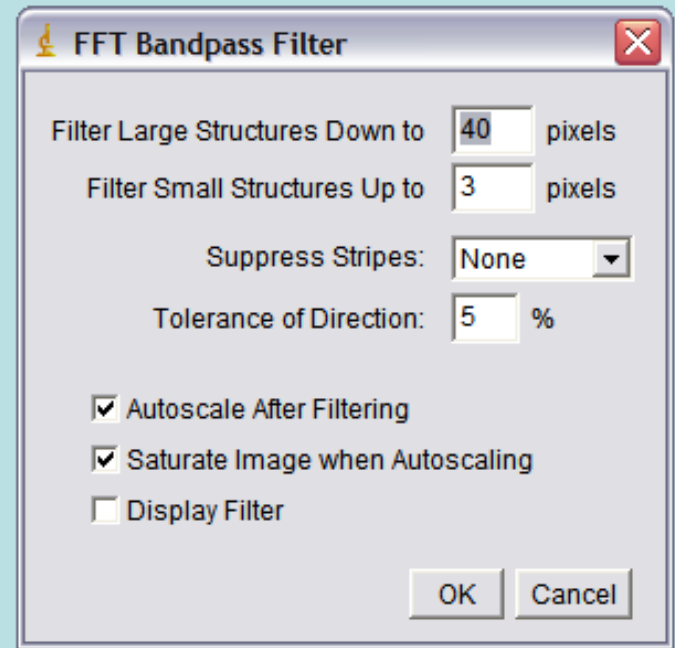
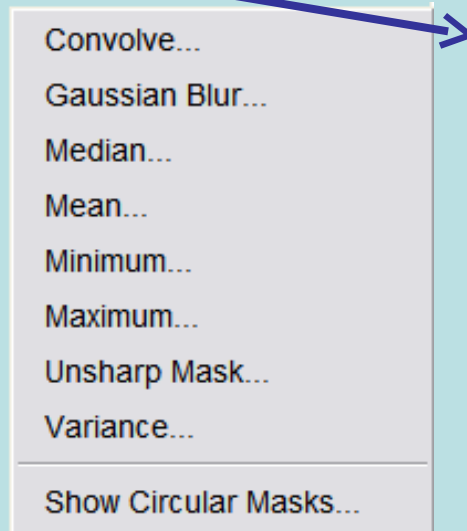
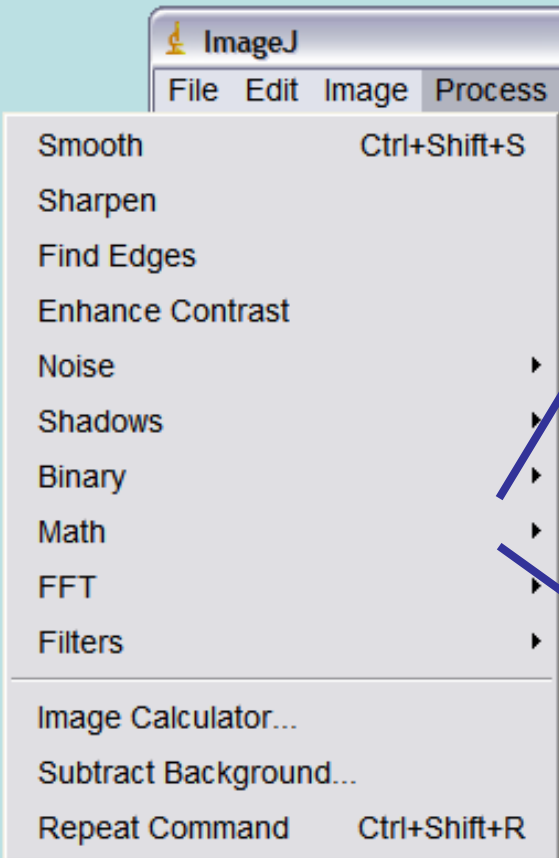
- Image
- Process
- Analyze
- Plugins
- Help



# Image Menu

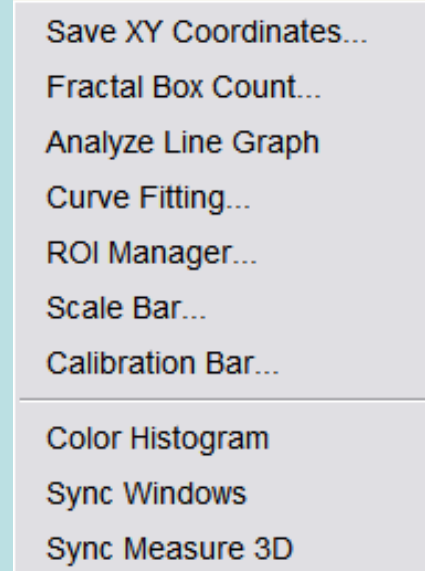
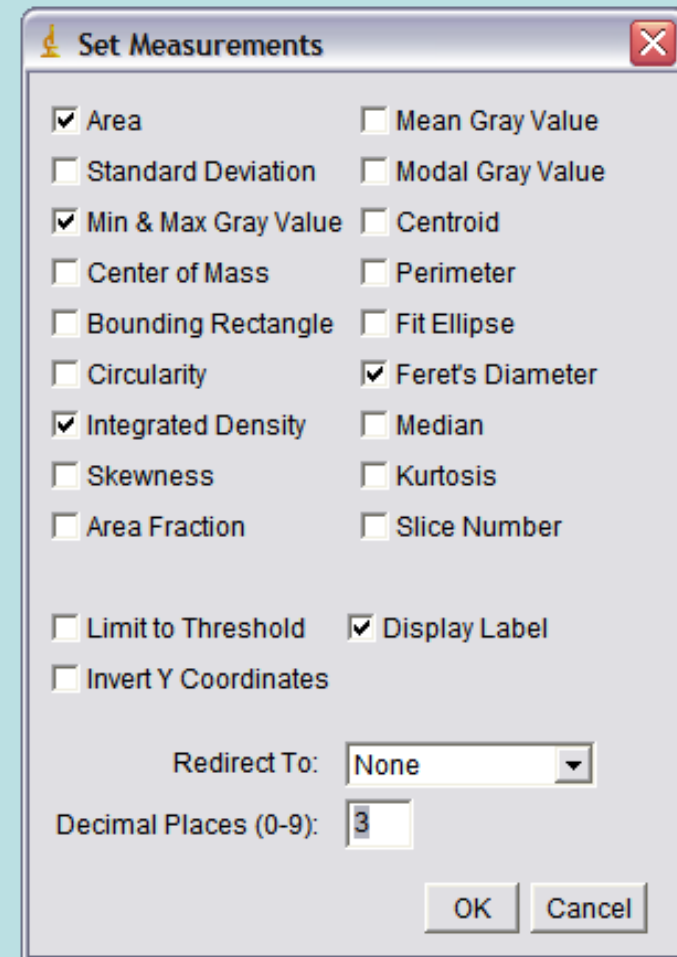
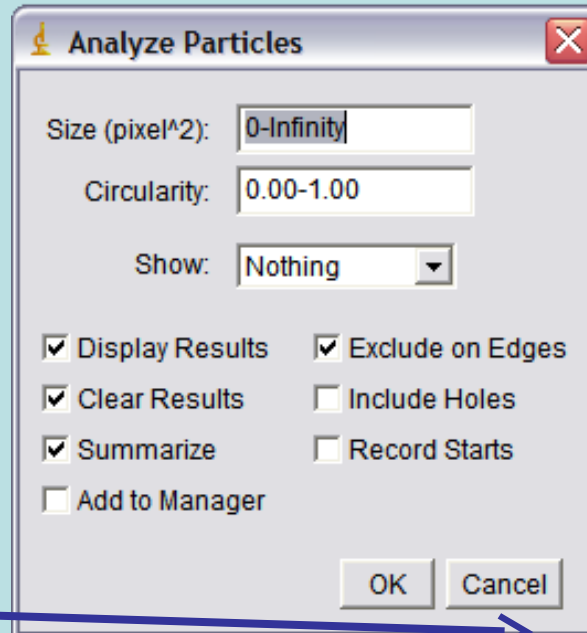
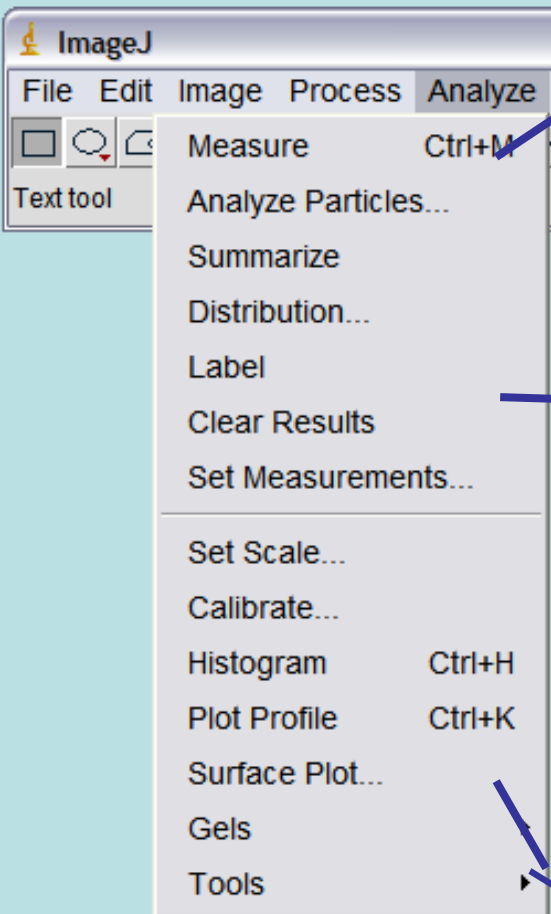


# Process Menu

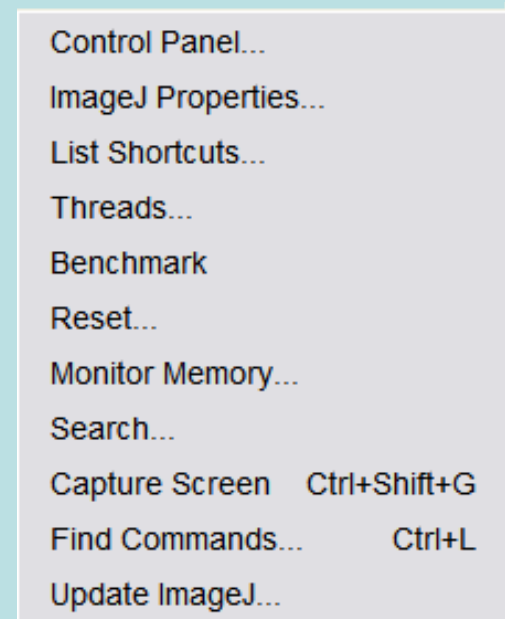
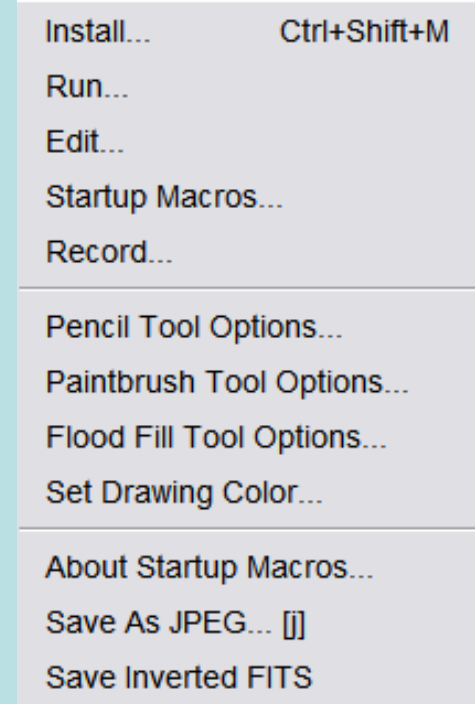
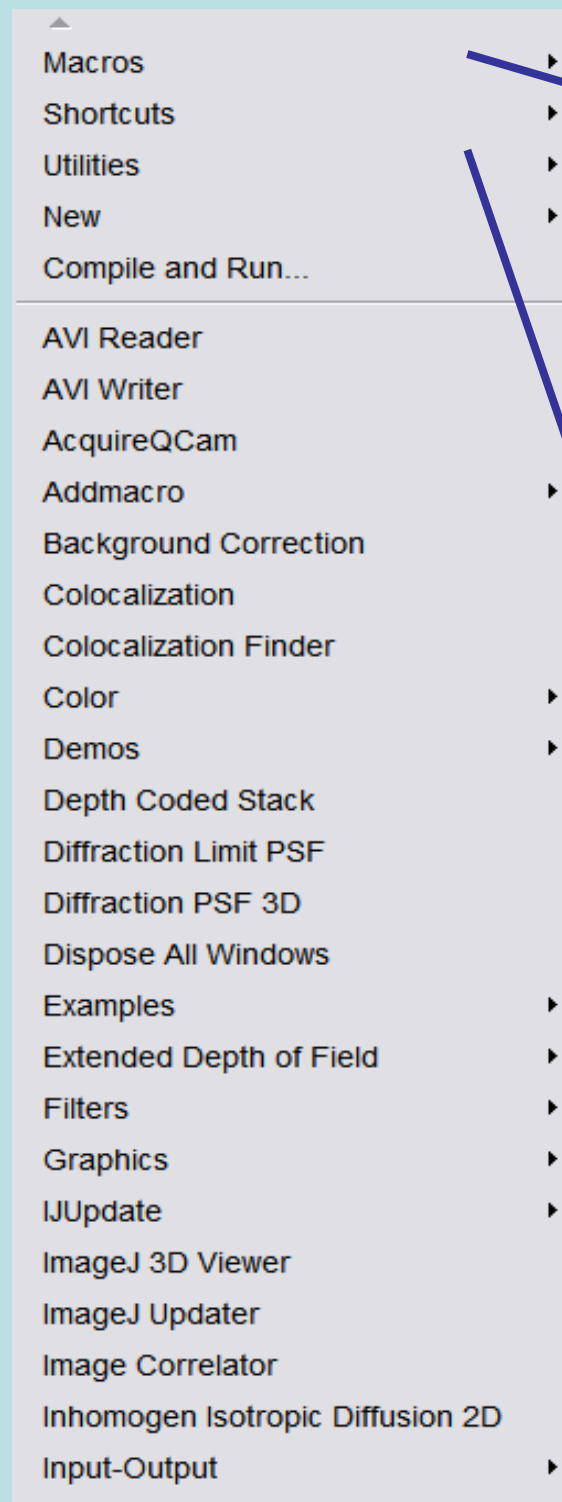




# Analyze Menu



# Plugins Menu





# FIJI

Fiji is an image processing package. It can be described as a distribution of ImageJ (and soon ImageJ2) together with Java, Java 3D and a lot of plugins organized into a coherent menu structure. Fiji compares to ImageJ as Ubuntu compares to Linux.

<http://fiji.sc/wiki/index.php/Fiji>



```
dir1 = getDirectory("Choose a Directory ");
```

```
list = getFileList(dir1);
```

```
ddir1 =dir1+"\\tiffs\\";
```

```
File.makeDirectory(ddir1);
```

```
for (i=0; i<list.length; i++) {
```

```
file = list[i];
```

```
file2 = dir1 + list[i];
```

```
open(file2);
```

```
//name1 = list[i] + " - C=0";
```

```
//name2 = list[i] + " - C=1";
```

```
//name3 = list[i] + " - C=2";
```

```
//selectWindow(name1);
```

```
run("Brightness/Contrast...");
```

```
setMinAndMax(0, 4095);
```

```
//selectWindow(name2);
```

```
run("Brightness/Contrast...");
```

```
setMinAndMax(0, 4095);
```

```
//selectWindow(name3);
```

```
setMinAndMax(0, 4095);
```

```
run("Brightness/Contrast...");
```

```
setMinAndMax(0, 4095);
```

```
//run("Concatenate...", "stack1=name1 stack2=name2 title=stack");
```

```
//run("Concatenate...", "stack1=name3 stack2=stack title=stacks");
```

```
name1 = ddir1 + i;
```

```
saveAs("Tiff", name1);
```

```
close();
```

## **IMAGEJ → Writing Macros**

*(use the macro-recorder under  
plugins—macro--record to get started)*

```
}
```



## IMAGEJ → Writing Macros

```
Dialog.create("TILER Z-Project")
numcols = 3;
numrows = 3;
Dialog.addNumber("ROWS", numcols);
Dialog.addNumber("COLUMNS", numrows);
Dialog.show();

numslices = nSlices;
numrows2 = numrows*numcols;
last=nSlices*numrows2;

run("Brightness/Contrast...");
setMinAndMax(0, 4095);
call("ij.ImagePlus.setDefault16bitRange", 0);
run("Make Montage...", "columns=numslices rows=numrows2 scale=1 first=1 last=last
increment=1 border=0 font=12");
run("Montage to Stack...", "images_per_row=numslices images_per_column=1
border=0");
run("Z Project...", "start=1 stop=numslices projection=[Max Intensity]");
run("Montage to Stack...", "images_per_row=1 images_per_column=numrows2
border=0");
run("Make Montage...", "columns=numcols rows=numrows scale=1 first=1
last=numrows2 increment=1 border=0 font=12");
```

```
dir1 = getDirectory("Choose a Directory ");
```

```
list = getFileList(dir1);
```

```
ddir1 =dir1+"\\sum\\";
```

```
File.makeDirectory(ddir1);
```

```
ddir2 =dir1+"\\max\\";
```

```
File.makeDirectory(ddir2);
```

```
for (i=0; i<list.length; i++) {
```

```
file = list[i];
```

```
file2 = dir1 + list[i];
```

```
open(file2);
```

```
run("Brightness/Contrast...");
```

```
setMinAndMax(0, 4095);
```

```
call("ij.ImagePlus.setDefault16bitRange", 0);
```

```
run("Stack Splitter", "number=2");
```

```
r = "(stk_0002_)" + file;
```

```
g = "(stk_0001_)" + file;
```

```
b = "(none)" + file;
```

```
selectWindow(file);
```

```
run("Close");
```

```
run("Merge Channels...", "red=r green=g blue=*None* gray=*None*");
```

```
run("Z Project...", "projection=[Sum Slices]");
```

```
name1 = ddir1 + "SUM_RGB" + file;
```

```
saveAs("Tiff", name1);
```

```
selectWindow("RGB");
```

```
run("Z Project...", " projection=[Max Intensity]");
```

```
name1 = ddir2 + "MAX_RGB" + file ;
```

```
saveAs("Tiff", name1);
```

```
close();
```

```
close();
```

```
close();
```

```
}
```

```
dir2 = dir1+"\\max\\";
```

```
list = getFileList(dir2);
```

```
for (i=0; i<list.length; i++) {
```

```
file = dir2 + list[i];
```

```
match = indexOf(file, "RGB");
```

```
file = list[i];
```

```
if (match > 0) {
```

```
open(dir2 + list[i]);
```

```
run("Split Channels");
```

```
selectWindow(file+ " (green)");
```

```
setThreshold(94, 255);
```

```
run("Convert to Mask");
```

```
selectWindow(file+ " (green)");
```

```
wait(2000);
```

```
run("Invert");
```

```
wait(1000);
```

```
selectWindow(file+ " (green)");
```

```
wait(1000);
```

```
imageCalculator("Subtract create", file + " (red)",file + " (green)");
```

```
run("Measure");
```

```
run("Close All");
```

# IMAGEJ → Writing Macros

```
selectWindow("Results");
```

```
name = dir2 + "Results.xls";
```

```
run("Text...", "save=name");
```

```
run("Clear Results");
```

```
//print(f, list[2] + "\t");
```

```
//selectImage(-3);
```

```
//saveAs("Raw Data", "./.raw");
```

```
//selectImage(-3);
```

```
//saveAs("Raw Data", "./.raw");
```

```
//selectImage(-4);
```

```
//saveAs("Raw Data", "./.raw");
```

```
dir2 = dir1+"\\sum\\";
```

```
list = getFileList(dir2);
```

```
for (i=0; i<list.length; i++) {
```

```
file = dir2 + list[i];
```

```
match = indexOf(file, "RGB");
```

```
file = list[i];
```

```
if (match > 0) {
```

```
open(dir2 + list[i]);
```

```
run("Split Channels");
```

```
selectWindow(file+ " (green)");
```

```
setThreshold(94, 255);
```

```
run("Convert to Mask");
```

```
selectWindow(file+ " (green)");
```

```
Wait(2000);
```

```
run("Invert");
```

```
wait(1000);
```

```
selectWindow(file+ " (green)");
```

```
wait(1000);
```

```
imageCalculator("Subtract create", file + " (red)",file + " (green)");
```

```
run("Measure");
```

```
run("Close All");
```

```
}
```

```
}
```

```
selectWindow("Results");
```

```
name = dir2 + "Results.xls";
```

```
run("Text...", "save=name");
```

```
run("Clear Results");
```

```
}
```



# IMAGEJ → Writing Plugins

```
import ij.*;
import ij.process.*;
import ij.gui.*;
import ij.plugin.*;
public class Demontager_ implements PlugIn {
    /** This plugin converts a montaged image to an image stack
        based on the number of rows and columns input by the user.
        This is the opposite of what the Image/Stacks/Make Montage
        command does.
    */
    public void run(String arg) {
        ImagePlus imp = IJ.getImage();
        GenericDialog gd = new GenericDialog("Demontager");
        gd.addNumericField("Columns:" , 1,0);
        gd.addNumericField("Rows:" , 1,0);
        gd.showDialog();
        if (gd.wasCanceled())
            return;
        int columns = (int)gd.getNextNumber();
        int rows = (int)gd.getNextNumber();
        ImageProcessor ip = imp.getProcessor();
        ImageStack stack = demontage(ip, columns, rows);
        new ImagePlus("Demontage", stack).show();
    }

    public ImageStack demontage(ImageProcessor ip, int columns, int rows) {
        int width = ip.getWidth();
        int height = ip.getHeight();
        int cropwidth = width/columns;
        int cropheight = height/rows;
        ImageStack stack = new ImageStack(cropwidth, cropheight);
        for (int j=1; j<=rows; j++) {
            for (int i = 1; i<=columns; i++) {
                int x = i * cropwidth - cropwidth;
                int y = j * cropheight - cropheight;
                ip.setRoi(x, y, cropwidth, cropheight);
                stack.addSlice(null, ip.crop());
            }
        }
        return stack;
    }
}
```

# Automating/ Calling ImageJ from other Programs

## *Call it from the command line*

### Plugins located in same directory:

```
java -cp ij.jar:. analyze blobs.tif > results.txt (Unix)
java -cp ij.jar;. analyze blobs.tif > results.txt (Windows)
```

### Or Define the Directory:

```
java -Dplugins.dir=/usr/local/ImageJ -cp /usr/local/ij.jar:. analyze blobs.tif > results.txt (Unix)
java -Dplugins.dir=C:\ImageJ -cp C:\ImageJ\ij.jar;. analyze blobs.tif > results.txt (Windows)
```

## *Run it in Batch mode → no GUI*

### *-batch path [arg]*

Runs a macro or script in batch (no GUI) mode, passing it an optional argument. ImageJ exits when the macro finishes.



# Confounding Factors in Data

- **Brightness inhomogeneities (laterally and axially (PSF) –confocal)**
- **Sample auto fluorescence**
- **Poorly/incompletely stained**
- **Transected structures due to physical sectioning**
- **Alignment and recombination (tiled data or badly aligned image stacks)**
- **Dense overlapping structures**
- **Sample variability**
- **Size**
- **Processing time**

**Error? – *confidence estimates***

# ALL PROJECTS

- **Implemented/useable as ImageJ plugin or macro**
  - *Call from command line?*
- **Extendable to other data**
  - *Test/works on all provided data*
- **automated**
  - *minimal user input to trigger*
- **Simple Graphical Interface**
  - **Sensible feedback of results**
    - *Non-destructive image overlay*
    - *Table of results/graphs*
- **Error warnings**
- ***Don't reinvent the Wheel!!!***



# The Data!

<http://drosophila.biology.kent.edu/users/rclement/resources/data/>

-multichannel data, not all channels necessary

-some operations require different channels

-*use 3D data (not 2D projections)*

## **1) Tiled data reconstruction and correction**

2-tiled data/cortex/test10\_\_20120411\_103225

2-tiled data/spinalcord/test10\_\_20120504\_093927

## **2) Astrocyte Cell classification**

3-astrocyte classification/single

*-only use astrocyte data channel*

3-astrocyte classification/multiple

*-astrocyte data for segmenting, nuclei for seed points*

## **3) Microvessel analysis**

4-microvessel analysis/big

4-microvessel analysis/small

# Imaging Projects – *Data examples*

*-show how to split channels  
-brightness adjust or reset*

## **1) Tiled data reconstruction and correction**

**-show projection image and MATL file**

## **2) Astrocyte Cell classification**

**-show projection image and sample data**

## **3) Microvessel analysis**

**-show projection image and channel data**