Workshop:
Image processing and analysis with ImageJ and MRI Cell Image Analyzer

Montpellier RIO Imaging

Volker Baecker

20.10.2008

Table of Contents

1. Introduction

In this workshop you will learn how to apply image analysis and processing techniques, using the public domain software ImageJ and some additions that have been added by Montpellier RIO Imaging. ImageJ has been written and is maintained by Wayne Rasband at the National Institute of Mental Health, Bethesda, Maryland, USA.
ImageJ is the successor of the Macintosh software NIH Image written by Wayne Rasband. ImageJ is written in Java, which means that it can be run on any system for which a java runtime environment (jre) exists. It can be run under Windows, Mac, Linux and other systems. It can be run as a browser plugin, on a website or as a standalone application. Because of the easy way in which ImageJ can be extended, using macros and plugins, a lot of functionality is available today, especially in the fields of microscopy and biology.

2. Installing ImageJ and MRI Cell Image Analyzer

2.1 Installing ImageJ

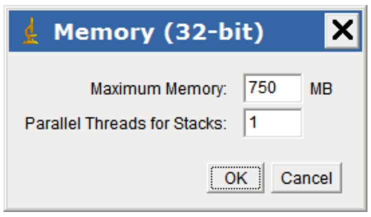
The ImageJ homepage is http://rsb.info.nih.gov/ij/. Go to the download page and download the appropriate version for your operating system. In this tutorial we will only use winXP. For windows there are two versions available:

- ImageJ bundled with the java runtime environment 1.6.0_05
- ImageJ without a java runtime environment

To use the last version the java runtime environment must already be installed on your computer. For this tutorial we download the first ImageJ bundle. Run the installer and follow the instructions on the screen. The first time ImageJ is started a configuration file for your installation will be created.
The installer will create a quick-start icon and a desktop icon. You can use these to start ImageJ. You can open images by dropping them on the desktop icon.

2.2 Memory Settings

Java applications will only use the memory allocated to them. Under *Edit>Options>Memory & Threads...*you can configure the memory available to ImageJ.

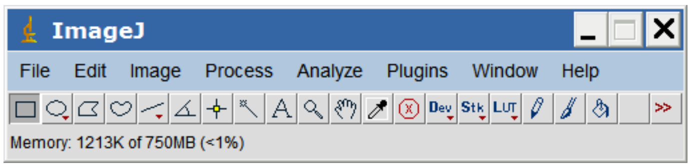


Maximum Memory: 750 MB
Parallel Threads for Stacks: 1

Illustration 1: The memory and threads settings of ImageJ

The maximum memory should be set to ¾ of the available memory on your machine. To find out how much memory your machine has, open the properties of *My Computer* and look for the amount of available RAM.
A dialog tells you that the change will be applied the next time you start ImageJ. The configuration is stored in the file *ImageJ.cfg* in the ImageJ folder. Should ImageJ not start

after you changed the memory settings, delete the file ImageJ.cfg, restart ImageJ and set the maximum memory to a lower value. A double-click on the lower part of the ImageJ launcher window displays the amount of used and available memory. You can use *Plugins>Utilities>Monitor Memory...*to monitor the memory usage.



Memory: 1213K of 750MB (<1%)

Illustration 2: A double click on the window displays memory information.

The number of *parallel threads for stacks* is by default set to the number of processors available in your system.

2.3 Upgrading ImageJ

To upgrade ImageJ, start ImageJ and go to *Plugins>Utilities>Update ImageJ...* Select the latest version from the list. ImageJ will be closed. Open it again and look at *help>about ImageJ* to see the current version number.
Alternatively you can download the latest pre-release version of ImageJ from http://rsb.info.nih.gov/ij/upgrade/
Download the file *ij.jar* and replace the version in the ImageJ home folder with the new version.

2.4 Associating file types with ImageJ

To open images by double-clicking we have to associate the file type with the ImageJ program. Download the zip-archive of example images from http://www.mri.cnrs.fr/mriwiki/uploads/images.zip and unzip the archive.
Right-click on a tif-image, select “open with” and “choose the program” from the context menu. On the dialog click the browse button and select *ImageJ.exe* from the ImageJ home directory. Check the “*always use this program to open files of this type*” option and click *ok* on the dialog. In the same way associate jpg-images with ImageJ.

2.5 Installing the “McMaster Biophotonics Facility” collection of plugins

The McMaster Biophotonics Facility offers an ImageJ manual and a collection of plugins. Using the collection of plugins guarantees that the user has the same setup that is described in the manual and that he has all plugins used in the manual available.
Download *mbf_plugins.zip* from http://rsb.info.nih.gov/ij/plugins/mbf-collection.html.
Extract everything from the zip file into the ImageJ home folder. When asked if files should be replaced answer always.
When you run ImageJ now you will get a warning due to the fact that some plugins are in two different subfolders of the plugins folder. Since the names of plugins containing an underscore are used as commands in ImageJ they must be unique. Delete one version of each plugin that exists two times and restart ImageJ.

2.6 Installing the Montpellier RIO Imaging plugins

We are going to use some plugins from Montpellier RIO Imaging, for :

- The visual scripting plugin
- The toolbox plugin
- The slide show control plugin
- The lookup table plugin

These are available at http://www.mri.cnrs.fr/index.php?m=38. To find them you can go to the homepage of Montpellier RIO Imaging: www.mri.cnrs.fr. In the English version click *Research and development>MRI-R&D-Software>MRI Cell Image Analyzer*. Follow the *Download MRI Cell Image Analyzer* link at the bottom of the page.
Download the two zip-archives mri-plugins-base.zip and mri-all-plugins.zip. To install them they must be extracted in the ImageJ home folder.
The AutoRun macro in the file StartupMacros.txt in the macros folder is run each time ImageJ is started. We can use it to switch to the MRI tool set at startup. Open the file from *Plugins>Macros>Startup Macros...*. Find the AutoRun macro. You might need to un-comment it by removing the // characters. Add the line

```
      call("gui.CellImageAnalyzer.switchToMRIToolset");
```

Save the changed StartupMacros.txt file.
Any menu command that can be found in the ImageJ launcher window can be used as an argument to the run command and any static java method can be called using the call command.

3. Help and Documentation

At the page http://rsb.info.nih.gov/ij/docs/index.html you can find help for the menu commands as well as a link to the MBF ImageJ manual. The page can be accessed from within ImageJ via the menu *Help>Documentation...*.

Further information can be found at the ImageJ Documentation Wiki:
http://imagejdocu.tudor.lu/imagej-documentation-wiki

The book “Digital Image Processing – An Algorithmic Introduction using Java” can be found here: http://www.imagingbook.com/index.php?id=15
Wilhelm BURGER • Mark J. BURGE
Digital Image Processing – An Algorithmic Introduction using Java
Textbook with 560 pages, 271 figures and 17 tables
© Springer 2008
ISBN: 978-1-84628-379-6

You can find more detailed explanation of some image processing techniques in the hyperlink image processing ressource (hipr) at:

http://homepages.inf.ed.ac.uk/rbf/HIPR2/index.htm

4. Display and image enhancements

4.1 What is a digital image

Before we start to work on images we should answer a simple question:
"What is a digital image?".
There are different kinds of images, of course. Images can for example be two or three dimensional and they can have one or more color channels.
In the simplest case a digital image is a two dimensional grid of width x height cells. Each cell is a square with sides of length p. The squares are called pixels and p is called the pixel size. To each pixel, an integer value between 0 and a maximum value is associated. These values are interpreted as intensities. 0 usually means no intensity, yielding a black pixel. Higher values usually mean lighter pixels and lower values mean darker pixels.
The word pixel is build from "pix element". Pix is a common abbreviation for picture.
Open the image *example1*. You can either double click the image or use File>open from the ImageJ window. When an ImageJ window is active you can use keyboard shortcuts to call commands from the ImageJ menus. To get the open dialog, click on an ImageJ window and press 'o'.

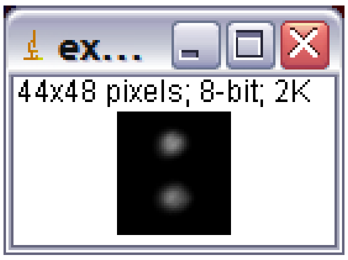


Illustration 3: The first example image.

Use the magnifying glass tool from the tool-box and click multiple times on the image to get the maximum zoom. You can now see the individual pixels.

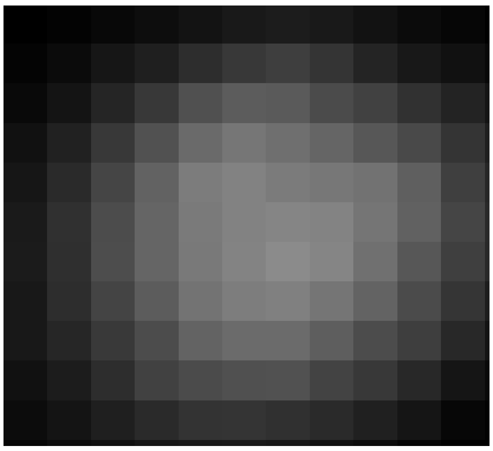
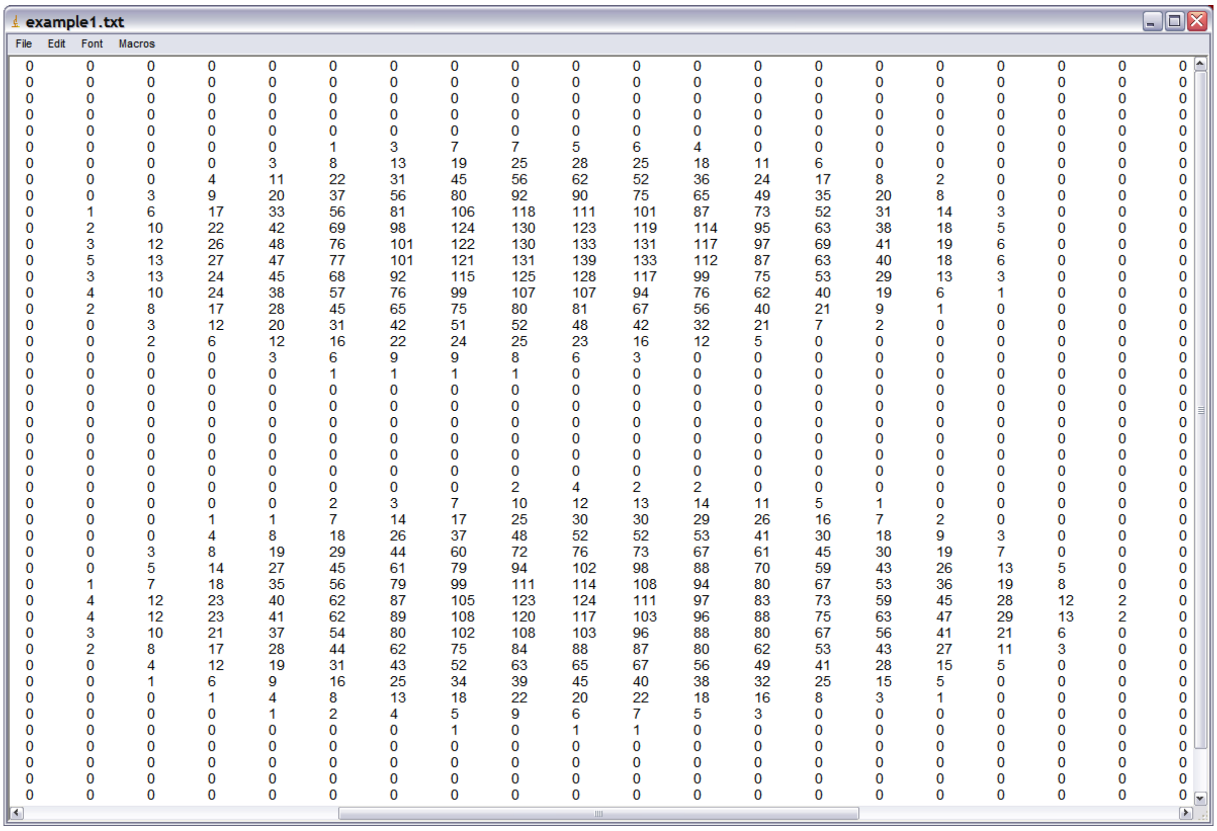


Illustration 4: With a higher zoom, individual pixels can be distinguished.

The coordinates and the intensity value of the pixel under the mouse pointer are displayed at the bottom of the ImageJ window. You can use the right mouse button to zoom out again. To scroll the image hold down space-bar, press the left mouse button and move the mouse. Remark that the origin of the coordinate system is in the upper left corner of the image with increasing coordinates from left to right and top to bottom.
Save the image as a text image to the desktop and open it with a text editor or drop it on the ImageJ window. You see the image as a matrix of intensity values.



```
0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
0     0     0     0     0     1     3     7     7     5     6     4     0     0     0     0     0     0     0     0
0     0     0     0     3     8     13    19    25    28    25    18    11    6     0     0     0     0     0     0
0     0     0     4     11    22    31    45    56    62    52    36    24    17    8     2     0     0     0     0
0     0     3     9     20    37    56    80    92    90    75    65    49    35    20    8     0     0     0     0
0     1     6     17    33    56    81    106   118   111   101   87    73    52    31    14    3     0     0     0
0     2     10    22    42    69    98    124   130   123   119   114   95    63    38    18    5     0     0     0
0     3     12    26    48    76    101   122   130   133   131   117   97    69    41    19    6     0     0     0
0     5     13    27    47    77    101   121   131   139   133   112   87    63    40    18    6     0     0     0
0     3     13    24    45    68    92    115   125   128   117   99    75    53    29    13    3     0     0     0
0     4     10    24    38    57    76    99    107   107   94    76    62    40    19    6     1     0     0     0
0     2     8     17    28    45    65    75    80    81    67    56    40    21    9     1     0     0     0     0
0     0     3     12    20    31    42    51    52    48    42    32    21    7     2     0     0     0     0     0
0     0     2     6     12    16    22    24    25    23    16    12    5     0     0     0     0     0     0     0
0     0     0     0     3     6     9     9     8     6     3     0     0     0     0     0     0     0     0     0
0     0     0     0     0     1     1     1     1     0     0     0     0     0     0     0     0     0     0     0
0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
0     0     0     0     0     0     0     0     2     4     2     2     0     0     0     0     0     0     0     0
0     0     0     0     0     2     3     7     10    12    13    14    11    5     1     0     0     0     0     0
0     0     0     1     1     7     14    17    25    30    30    29    26    16    7     2     0     0     0     0
0     0     0     4     8     18    26    37    48    52    52    53    41    30    18    9     3     0     0     0
0     0     3     8     19    29    44    60    72    76    73    67    61    45    30    19    7     0     0     0
0     0     5     14    27    45    61    79    94    102   98    88    70    59    43    26    13    5     0     0
0     1     7     18    35    56    79    99    111   114   108   94    80    67    53    36    19    8     0     0
0     4     12    23    40    62    87    105   123   124   111   97    83    73    59    45    28    12    2     0
0     4     12    23    41    62    89    108   120   117   103   96    88    75    63    47    29    13    2     0
0     3     10    21    37    54    80    102   108   103   96    88    80    67    56    41    21    6     0     0
0     2     8     17    28    44    62    75    84    88    87    80    62    53    43    27    11    3     0     0
0     0     4     12    19    31    43    52    63    65    67    56    49    41    28    15    5     0     0     0
0     0     1     6     9     16    25    34    39    45    40    38    32    25    15    5     0     0     0     0
0     0     0     1     4     8     13    18    22    20    22    18    16    8     3     1     0     0     0     0
0     0     0     0     1     2     4     5     9     6     7     5     3     0     0     0     0     0     0     0
0     0     0     0     0     0     0     1     0     1     1     0     0     0     0     0     0     0     0     0
0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
```

Illustration 5: The image as a matrix of intensity values.

4.2 Brightness and Contrast adjustment

Open the image *bc-adjust.tif*. The image is dark and the contrast is very low. Open the Brightness and Contrast Adjuster from the menu *Image>Adjust>Brightness/Contrast* or press *Shift+C* or run the B/C-Adjuster from the Lookup Tables tool set.

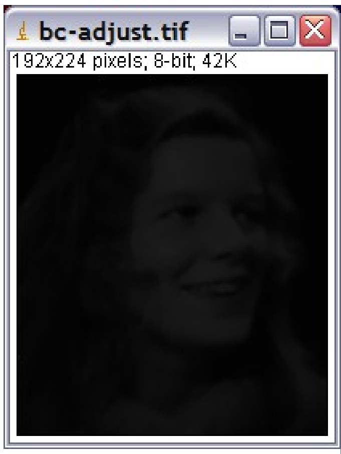


Illustration 6: The intensities are to low to distinguish details in the image.

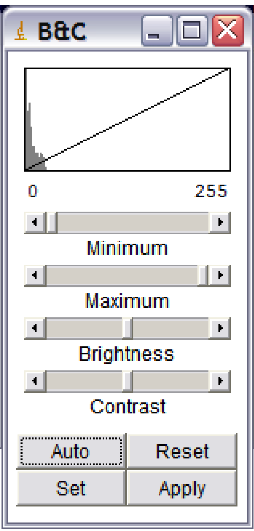


Illustration 7: The brightness and contrast adjuster.

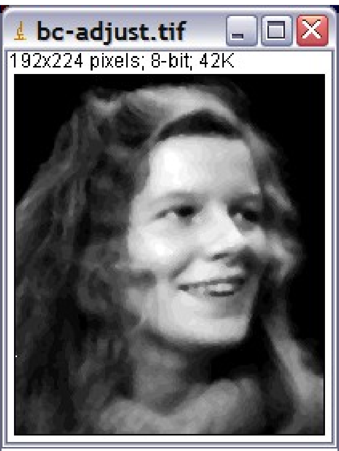


Illustration 8: The image after the auto-adjustment.

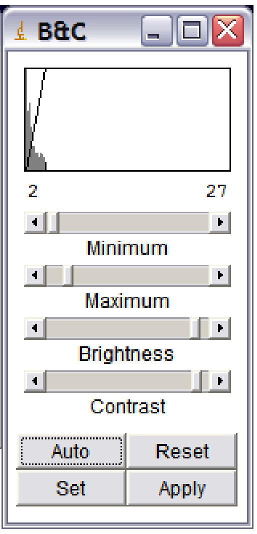
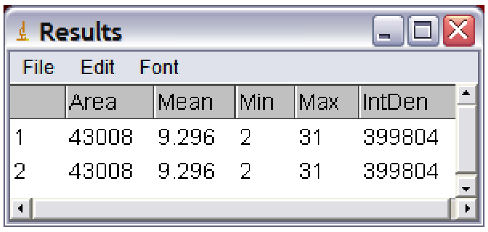


Illustration 9: After the adjustment intensities below 3 will be displayed as black and intensities above 26 as the 255.

Press the *Auto* button to make an automatic adjustment. With *Reset* you can go back to the original display. Use the four sliders to adjust brightness and contrast manually. Remark that only the displayed intensities are changed, the pixel values remain unchanged.
Press the measure button on the toolbox or select the image and press *Ctrl+m*, then change the brightness/contrast and measure again. Compare the values *mean* and *IntDen*, which give the average and total intensity in the image. If you can't find *mean* or *IntDen* in the results table, open the dialog *Analyze>Set Measurements...* and check *Mean Gray Value* and *Integrated Density*.

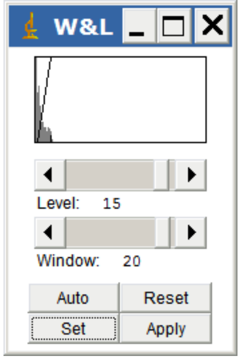


	Area	Mean	Min	Max	IntDen
1	43008	9.296	2	31	399804
2	43008	9.296	2	31	399804

Illustration 10: The ImageJ results table.

How does the adjustment work? When the image is displayed two points are stored together with the image. The point (*min*, 0) and the point (*max*, *maxIntensity*). All intensity values below *min* are displayed with the intensity zero, all intensity values above *max* will be

displayed with the intensity value *maxIntensity*. Values between *min* and *max* are mapped by the line equation that is defined by the two points (*min*,0) and (*max*, *maxIntensity*).
In the beginning when we opened the example image, *min* was 0 and *max* was 255 and the *maxIntensity* is 255. Since this defines a line with the gradient 1, each intensity value is mapped to itself. After we applied the auto adjustment, *min* becomes 2 and *max* 27, so pixels that have the intensity 27 are displayed with the value 255, pixels that have the intensity 20 with the value 184, and so on.
Now look at the image in the upper part of the B&C window. What you see there is the histogram of the image. On the x-axis are the intensity values from 0 to 255. On the y-axis you find the count of pixels in the image that have the intensity x. We'll come back to the histogram later when we talk about thresholding. For the brightness and contrast adjustment it shows you how many pixels you set to the max intensity and to zero. On the left you find the *min* display value and on the right the *max* display value. The line that is drawn in the histogram shows the mapping of the values between *min* and *max*.
If you move the brightness slider to the right the whole line will move to the left and lower intensity values will be mapped to higher display values, making the display brighter.
If you move the contrast slider to the right the gradient of the line increases so the intensity values closer together will be mapped to display values with a bigger distance.
You can use the set button to enter the min and max value directly. If you press apply, the pixel intensity values in the image will be changed to the values displayed.
The auto adjustment will look at the histogram of the image and adjust the min and max values in a way that a small percentage of the pixels become zero and *maxIntensity*. Sometimes this doesn't yield the desired results, especially when the object is very small compared to the background. In this case you can make a selection on the image and the auto adjuster will use the histogram of the selection to compute the *min* and *max* values that are then applied to the whole image.
Use the rectangular selection tool from the toolbox to make a selection on the image and press the auto button. Try this for different regions in the image.



Level: 15

Window: 20

Another way to adjust the display is the *Window&Level adjuster*. In this case you choose a middle value (the level) and a range around the middle value (the window). If you choose for example *level* 15 and *window* 24 you get a *min* value of 3 and a *max* of 27.

4.3 Non linear display-adjustments

In the brightness and contrast adjustments in the last chapter, intensity values are mapped to display values by a linear function, meaning small intensities are changed by the same value as big values. In cases where very high and very low intensities should become visible a non-linear display adjustment is needed, so that small values can become bigger without saturating already big values. Strictly speaking non-linear display adjustments are not possible in ImageJ. But we can use a non-linear function to modify the intensity values in the image.

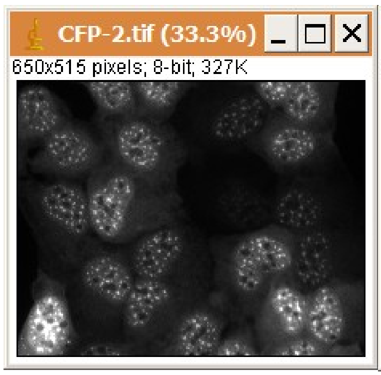


Illustration 11: The low intensities are invisible in the original image.

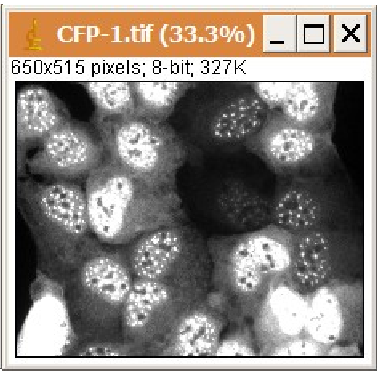


Illustration 12: A linear display adjustment makes low intensities visible but saturates high intensities.

4.3.1 Gamma correction

Instead of a linear function we will map intensities to display values using a function of the form $V_{out} = V_{in}^{\gamma}$.

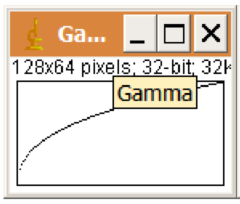


Illustration 13: gamma = 0.4

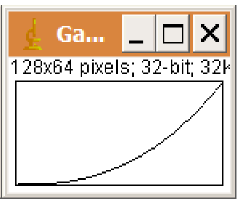


Illustration 14: gamma = 2.5

Use the gamma-scrollbar from the menu *Plugins>Utilities>Gamma scroll-bar* to apply a gamma correction on the image *cells.tif* or *cells2.tif*.

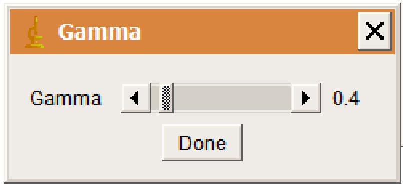


Illustration 15: The gamma-scrollbar.

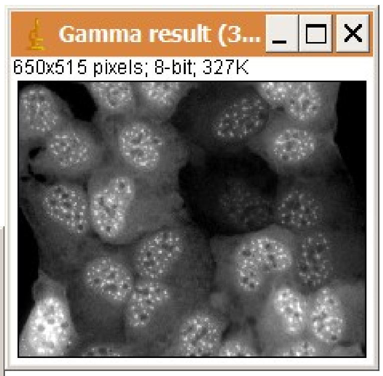
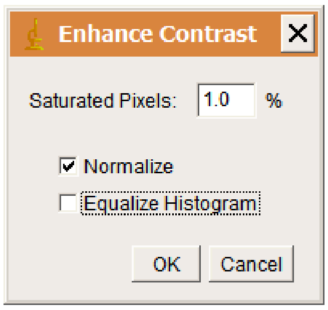


Illustration 16: The gamma-correction has been applied to the image.

4.3.2 Enhance Contrast

The enhance contrast tool, that can be found under *Process>Enhance Contrast*, has three different functions. When neither Normalize nor Equalize is selected, the display is adjusted in a way, that the given number of pixels becomes saturated. In this case the intensity values in the image are not changed.
When Normalize is selected, a contrast stretching or normalization is done.
The third function is the histogram equalization.



Enhance Contrast

Saturated Pixels: 1.0 %

[✓] Normalize
[ ] Equalize Histogram

Illustration 17: The enhance contrast dialog.

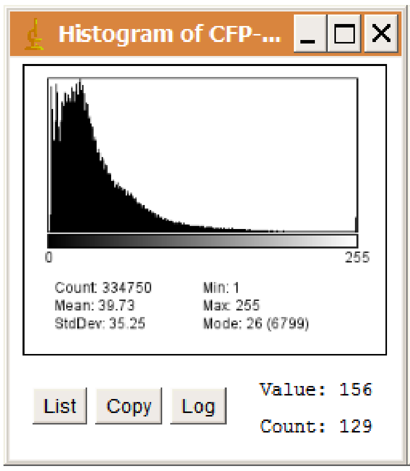
4.3.2.1 Normalization or contrast stretching

The normalization stretches the range of intensity values, so that the given percentage of pixels becomes saturated. After a normalization the min and max values for the given image type will be present in the image. If for example the minimum and maximum value in an image are 10 and 20 the values can be mapped to use the whole range between 0 and 255 by:

*i' = (i-image_min / (image_max – image_min)) * range_max*

*i'(10) = (10-10) / (20-10) * 255 = 0*
*i'(20) = (20-10) / (20-10) * 255 = 255*
*i'(15) = (15-10) / (20-10) *255 = 127.5*

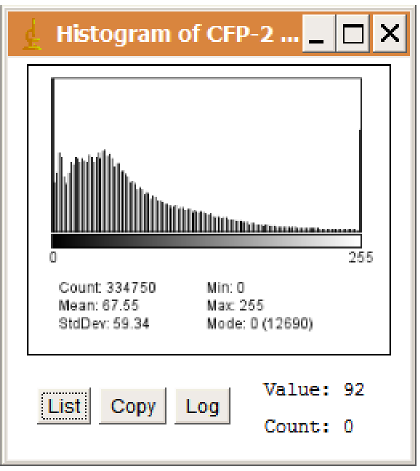
However this would be very sensible to outliers. A better way is to determine *image_min'* and image_max' so that a given percentage of pixels is below image_min' and above *image_max'*.



```
Count: 334750      Min: 1
Mean: 39.73        Max: 255
StdDev: 35.25      Mode: 26 (6799)

Value: 156
Count: 129
```

Illustration 18: Histogram before normalization.



```
Count: 334750      Min: 0
Mean: 67.55        Max: 255
StdDev: 59.34      Mode: 0 (12690)

Value: 92
Count: 0
```

Illustration 19: Histogram after normalization.

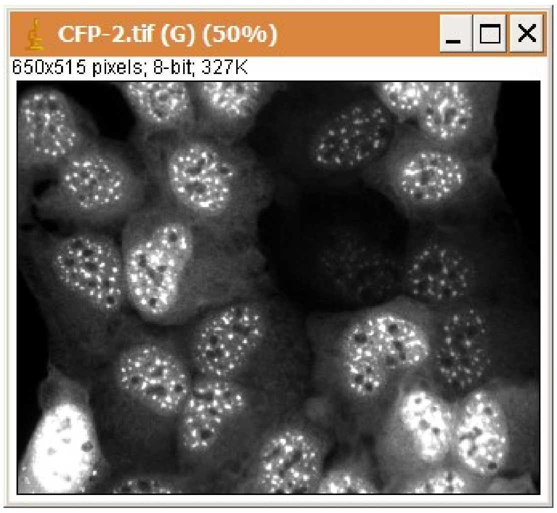


Illustration 20: The image after histogram normalization.

Using the enhance contrast tool with the normalize option has the same effect as using it without the normalize option and then pressing apply on the brightness and contrast adjuster.

4.3.2.2 Histogram Equalization

In the histogram equalization each intensity value *i* is replaced with the the sum of the histogram values for all intensities up to *i* (including *h(i)* itself). The result is then normalized to the *min* and *max* intensities of the image type. If you check equalize histogram in the enhance contrast dialog, the other input in the dialog is ignored. If you hold down alt, the classical histogram equalization will be executed. Otherwise a modified version, using the square root of the histogram values will be used. If a selection exists on the image, the calculation will be based on the histogram of the selection only.

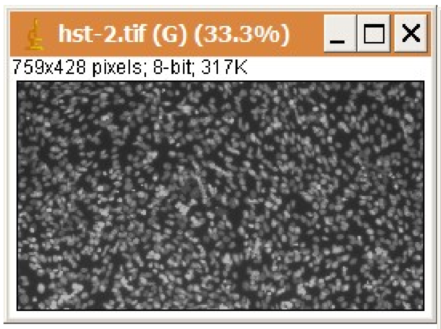
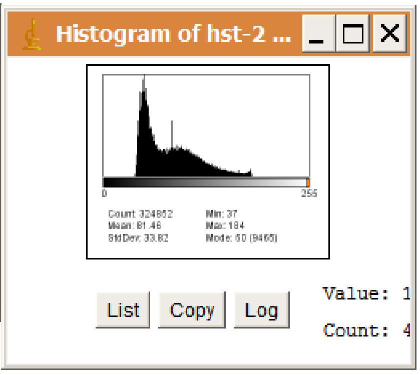


Illustration 21: The input image.



```
Count 324852     Min: 37
Mean: 81.46      Max: 184
StdDev: 33.82    Mode: 50 (9465)
```

Illustration 22: The histogram of the input image.

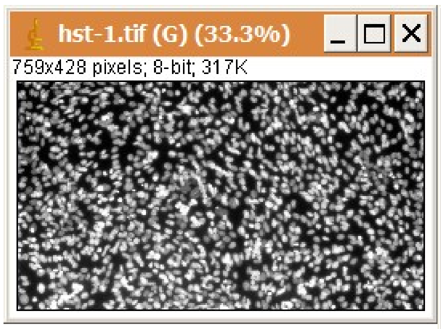
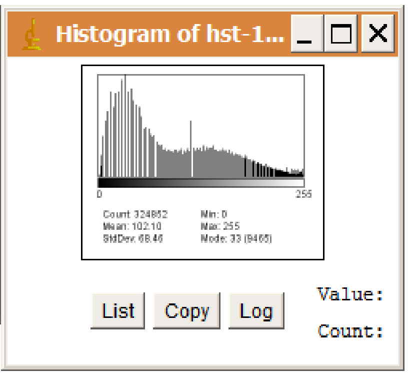


Illustration 23: The image after histogram equalization in the square root version.



```
Count 324852     Min: 0
Mean: 102.10     Max: 255
StdDev: 68.46    Mode: 33 (9465)
```

Illustration 24: The histogram of the image after histogram equalization in the square root version.

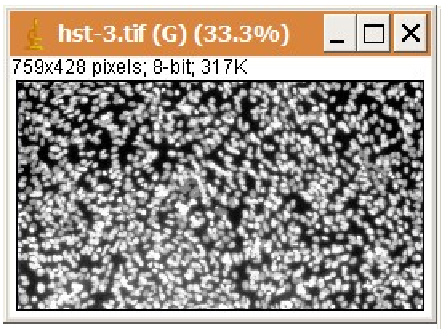
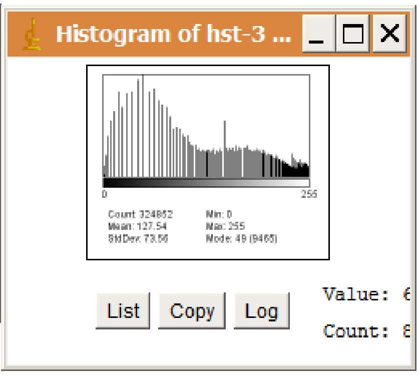


Illustration 25: The image after histogram equalization in the standard version (alt key down).



```
Count 324852     Min: 0
Mean: 127.54     Max: 255
StdDev: 73.56    Mode: 49 (9465)
```

Illustration 26: The histogram of the image after histogram equalization in the standard version (alt key down).

Histogram equalization augments the local contrast and is most useful when objects and background contain high and low intensities, as for example in brightfield microscopy. Compare the results of a linear adjustment a gamma adjustment and the histogram equalization on the image *cells2.tif*.
Try the histogram equalization on the image *roots.tif*.

4.4 Changing the palettes

We will now change the display of our image by mapping the intensity values to colors. This is done by the use of lookup tables. A lookup table is a table that maps the 255 intensity values to 255 arbitrary colors. Since colors are expressed by the proportion of the three basic colors red, green and blue, an entry that would map the intensity 255 to white would for example be:
255 -> 255, 255, 255
To map the intensity 0 to red the lookup-table entry would be:
0 -> 255, 0 , 0
Select the Lookup tables tool set. Click on the LUT button and open the Rainbow RGB lut. You see an image that shows the mapping of the intensity values to the colors. Each intensity value is represented by a column with the width of 1 pixel, painted in the corresponding color. Can you tell in which color the intensity value 200 is displayed?

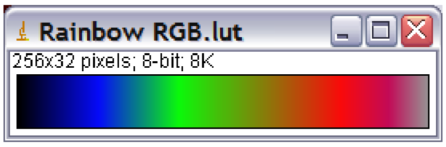


Illustration 27: A graphical representation of the lookup-table.

Go to the menu image>color>show lut. Above the color table you see how each color of the lut is mixed from the red, green and blue components.

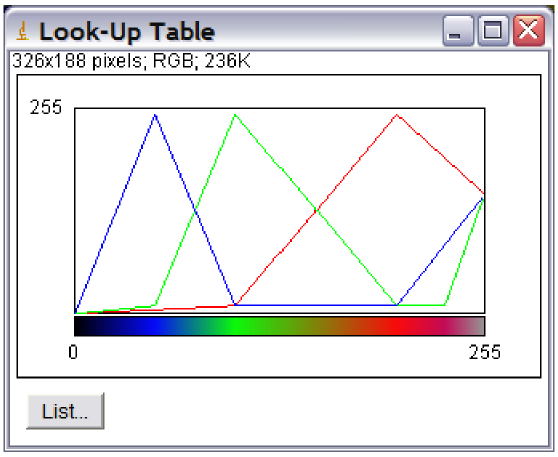
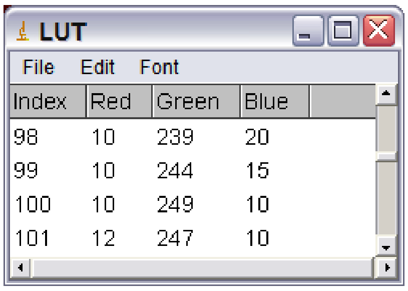


Illustration 28: The red, green and blue components for each index of the lookup-table.

If you click on the list button, you get the lookup-table in numerical form. What are the RGB-components of the intensity 100?



Index	Red	Green	Blue
98	10	239	20
99	10	244	15
100	10	249	10
101	12	247	10

Illustration 29: Part of the lookup-table.

Open the image cells and click on Rainbow RGB lut again. When images are opened the lookup-table will be applied to the active image. Try different lookup tables.

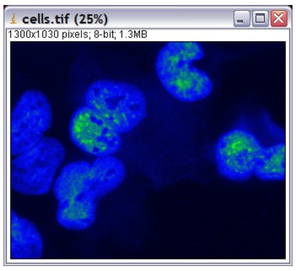


Illustration 30: The rainbow RGB lookup-table applied to an image.

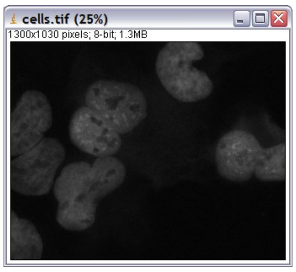


Illustration 31: The grey lookup-table applied to an image.

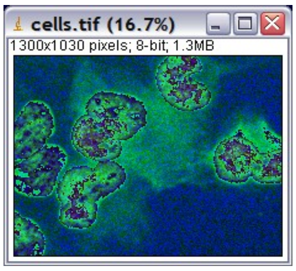


Illustration 32: The 3-3-2 RGB lookup table applied to an image.

The lookup table hilo is useful to adjust the display of our image. 0 will be displayed in blue, 255 in red and values in between will be displayed in gray. Can you describe how this lookup table looks like?

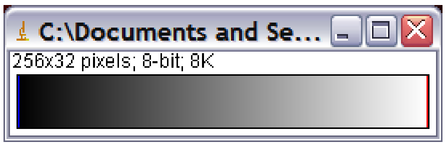


Illustration 33: The hilo lookup-table.

Apply it to the image and adjust the brightness and contrast in a way that most of the background is zero and only a small portion of pixels becomes saturated.

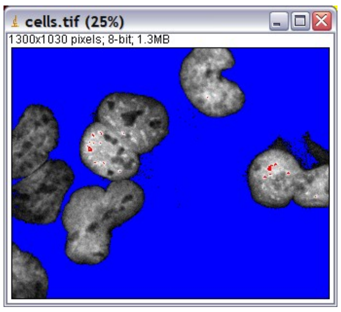


Illustration 34: Use the hilo lut to adjust brightness and contrast.

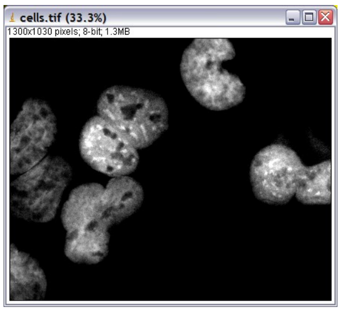


Illustration 35: The brightness and contrast adjustment has been applied and the lut has been changed back to grays.

Go to *image>color>edit>lut* or press the *Edit LUT* button from the Lookup Tables tool set. Change the lookup table in a way that 0 will be displayed in green, 255 in yellow and values between 100 and 120 in different shades of red, becoming lighter with higher intensity. Save the lookup table under a new name, apply it to the image and adjust the *B&C* again.

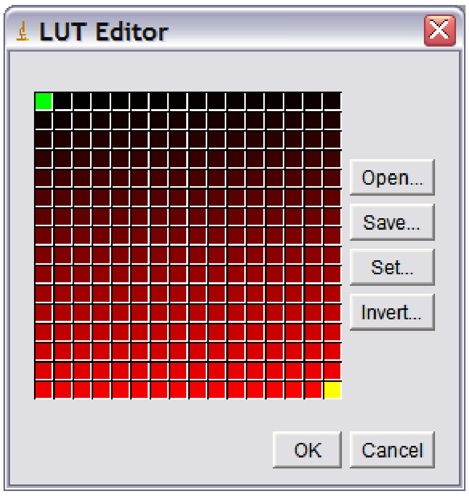


Illustration 36: The lut editor.

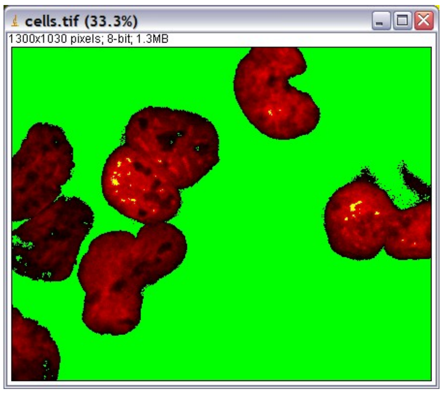


Illustration 37: The newly created lut applied to an image.

Open the *MRILookupTableTool* under *Plugins>Montpellier RIO Imaging*. The tool allows you to see the lookup tables and to apply them either to the active image or to all open images. Remark that the external lookup tables listed in the tool must be in the folder *_lut*.

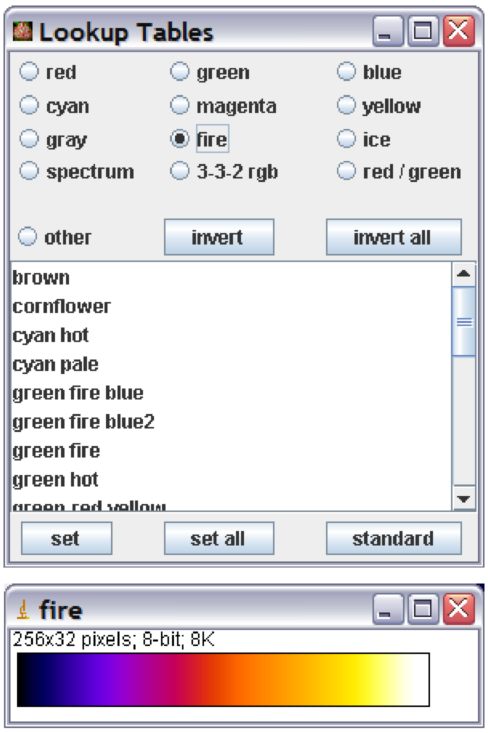


Illustration 38: The MRI look-up table tool.

Another convenient way to access all lookup tables is to tear off the *Image>Lookup Tables* menu by using *Plugins>Utilities>Control Panel*.

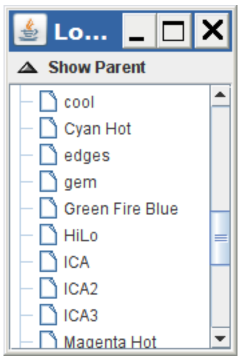


Illustration 39: Menus can be kept on the screen using the Control Panel.

4.5 Overlay of multiple channels

A common task in fluorescent microscopy is to create a combined image from the different channels. The task consists in creating an overlay of the different channels, in adjusting the display of each channel and eventually in transferring the settings from one overlay to another, to allow a visual comparison. In ImageJ this can be accomplished using so called hyperstacks. Hyperstacks allow to work with multidimensional images. The different dimensions are the x,y and z axis, the time and the channel (representing the color or wavelength).

Open the images *dapi 3.tif* and *rhod 3.tif* from the folder *11 – overlay.*

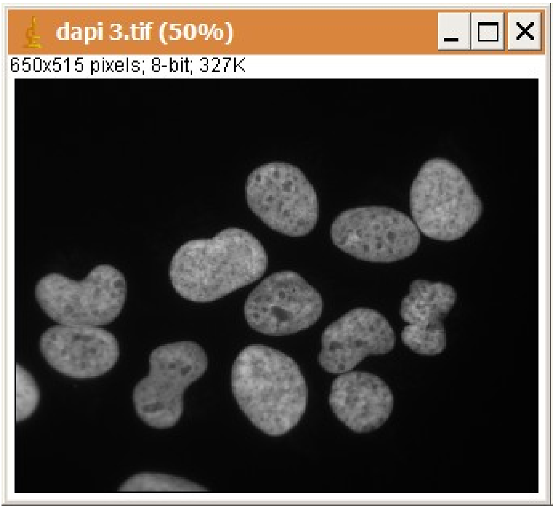


Illustration 40: The first channel contains the dapi staining.

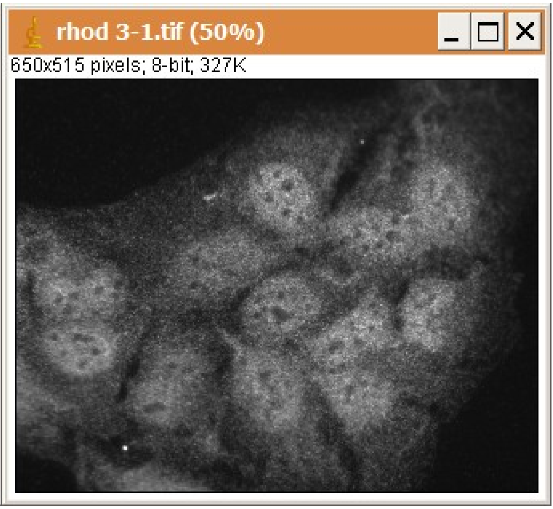
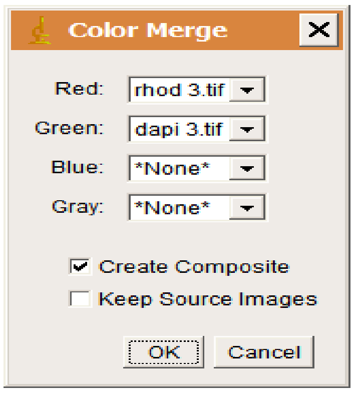


Illustration 41: The second channel contains the rhodamine staining.

Run the *Merge Channels...* command from the menu *Image>Color*.



Red: rhod 3.tif
Green: dapi 3.tif
Blue: *None*
Gray: *None*

☑ Create Composite
☐ Keep Source Images

Illustration 42: Dialog to create hyperstacks, i.e. multidimensional images.

Select *Create Composite* and press *ok*. You now see an overlay of the two channels. Open the Channels dialog from the menu *Image>HyperStacks* and the Brightness and Contrast Adjuster by pressing *shift+c* on the image. With the slider on the bottom of the stack window you select the channel to manipulate. If you move it to the right the histogram in the Brightness and Contrast tool becomes green indicating that the green channel is selected. If you move the slider back to the left the histogram becomes red again. You can change the

colors by applying lookup tables from the *LUT menu*. Adjust the display of the two channels using the *B&C* tool. Remark that the display adjustment of one channel remains when you change to the other channel. You don't need to press the apply button.
You can save a hyperstack and reload it without loosing the display adjustment. Save it in tif format using *File>Save* or *File>Save As*. Close the hyperstack window and reload the saved file using *File>Open*.
If you want to create a snapshot of the overlay click on the *More* button in the Channels window (*Image>Hyperstacks>Channels Tool...*) and select *Convert to RGB*.

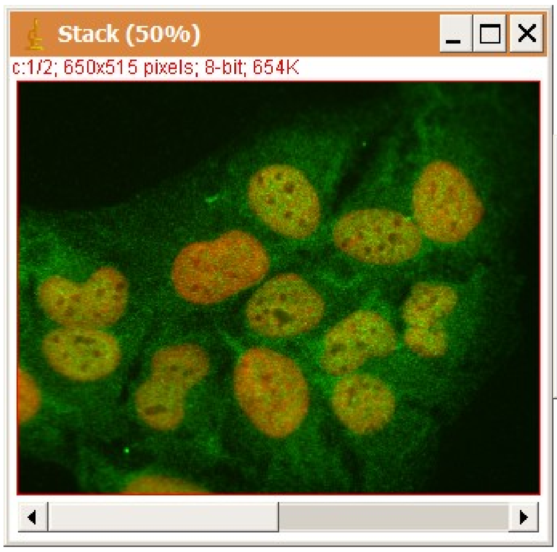


Illustration 43: The overlay of two channels.

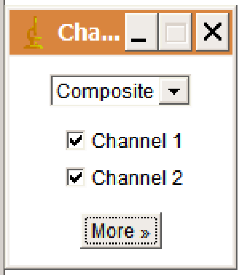


Illustration 44: The channel dialog allows to show and hide selected channels.

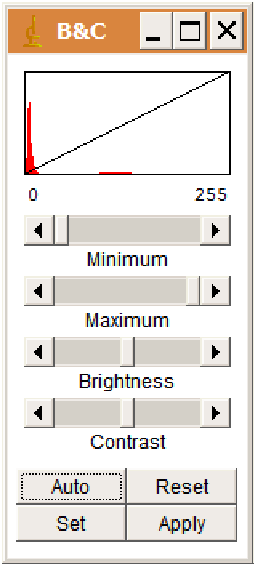


Illustration 45: The B&C tool works on the selected channel and displays the histogram in the color of the selected channel.

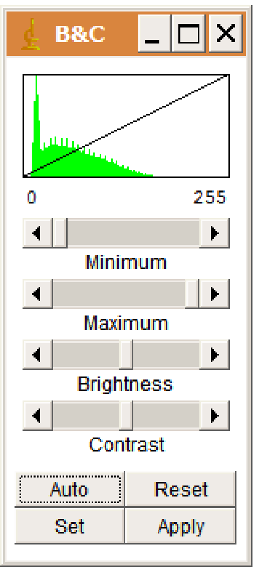
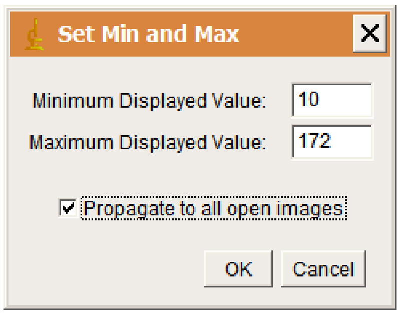


Illustration 46: Changing the selected channel automatically changes the channel on which the B&C tool works.

Create a second hyperstack from the images *dapi 5* and *rhod 5*. Imagine that you want to compare the intensities in the two images. To be able to do this you need to set the same display adjustments for both images. Select the red channel in both images. Adjust the display for the first image (for example by pressing reset followed by auto), then press the button set.

Select *Propagate to all open images* and press *ok*. The display of the red channel in the second image has been adjusted in the same way as the one in the first image now. Select the green channel in both images and repeat the procedure.


Set Min and Max

Minimum Displayed Value: 10

Maximum Displayed Value: 172

[✓ Propagate to all open images]

Illustration 47: The display settings can be propagated to all open images (same channel).

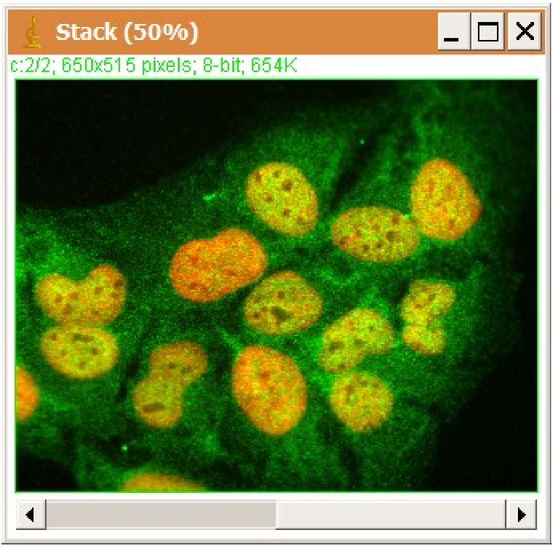


Illustration 48: The display settings have been optimized for the first image.

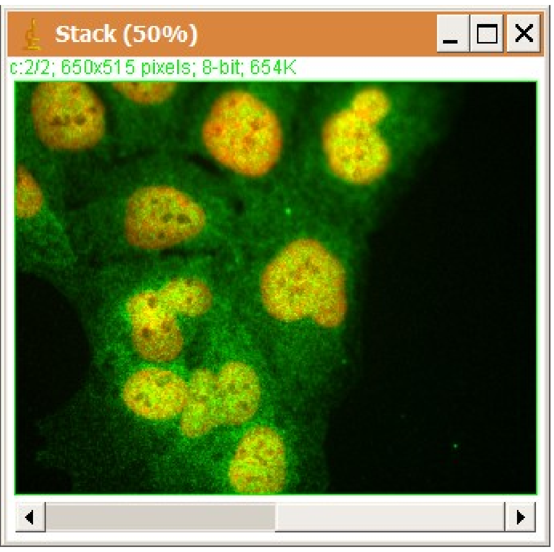


Illustration 49: The display settings have been transferred to the second image.

Create an overlay from the three images *b2RFP_gemDeltaC2_blue.tif*, *b2RFP_gemDeltaC2_green.tif* and *b2RFP_gemDeltaC2_red.tif* and adjust the display.

4.5.1 Aligning channels

You can use the composite display to align two channels. Open the images dapi 4.tif and Rhod 4.tif and create a hyperstack. As you can see the to channels have been shifted against each other. Use ctrl+a to select the image with the first channel selected, then press ctrl+x to cut the channel and ctrl+v to paste it again. You can now move the image of the first channel around. Correct the alignment then select a rectangle around the not empty area of the image and call Image>Crop to get rid of the border.

4.5.2 Overlay of volume images

We will now display an image with 54 z-slices and 3 channels. Open the images Dapi.stk, the Gtub.stk and the Actine.stk. Create the Composite image using *Image>Color>RGB Merge...* You now have a window with two sliders, one to select the channel and one to select the slice in the stack.

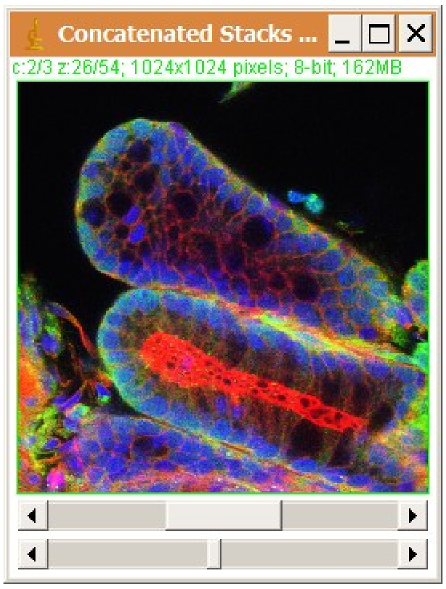
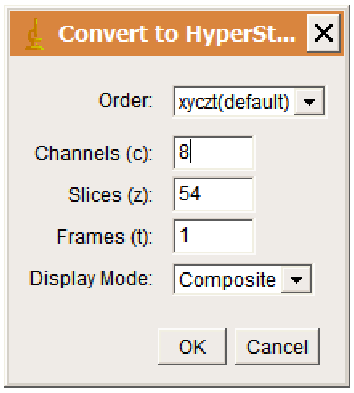


Illustration 50: A hyperstack of a volume image with three channels.

Two create a composite with more than four channels you can use the command *Image>Hyperstacks>Stack to Hyperstack.*



Convert to HyperSt...

Order: xyczt(default)
Channels (c): 8
Slices (z): 54
Frames (t): 1
Display Mode: Composite

Illustration 51: With the Stack to Hyperstack command hyperstacks with more than 4 channels can be created.

4.6 Noise suppression

Open the image *plant-noise.tif*. The image contains a high level of noise. Zoom into the image. The background should be homogeneous, but it contains a random distribution of intensities. The same random distribution is added to the signal.

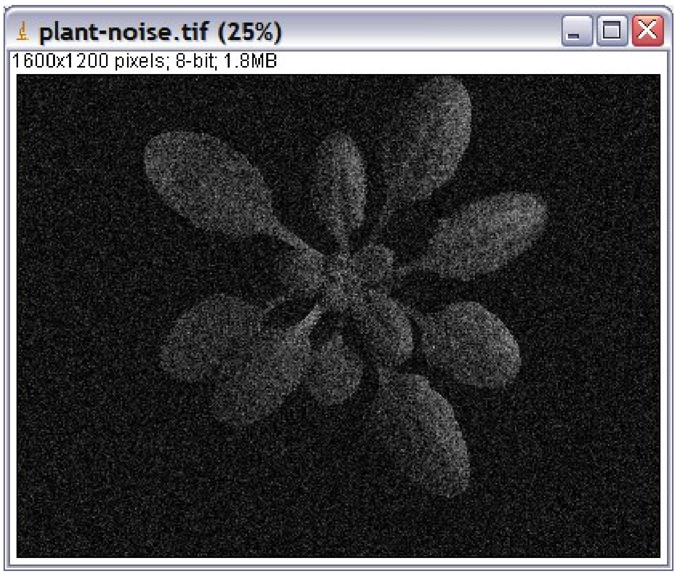


Illustration 52: *An image with a high level of noise.*

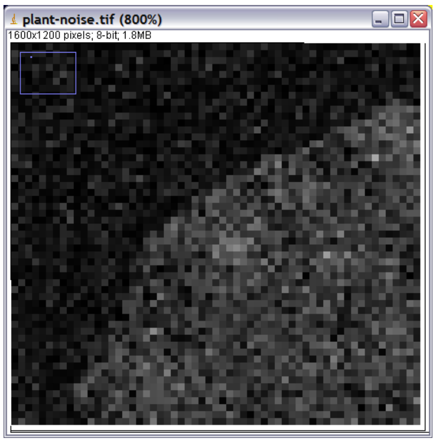


Illustration 53: Pixel values in background and object are changed in a random manner.

There are different possible sources of noise in an image. One kind of noise stems for example from the digital camera. In the camera incoming photons are transformed into an electrical charge by a charge coupled device or CCD. However some electrons are created within the CCD randomly. Another source of noise is the random emission of photons from your specimen, which is a quantum physical phenomenon.
The noise can pose a problem for the analysis of the image, for example in the separation of the objects from the background. If we have a light object on a dark background we could for example look for the lowest intensity value in the object and separate the object from the background by searching all pixels that have an intensity value higher than this threshold value. However the noise has changed the intensities in the background and in the object, so that there are very high intensities within the background and very low intensities in the object. We might have to pre-process our image to reduce the disturbing effect of noise.

4.6.1 Convolution filter

4.6.1.1 Mean filter

An evident way to smooth our image and to reduce the impact of the noise is to replace each pixel in the image by an average of the intensities in its neighborhood. This is called a mean filter in ImageJ. You can find it under Process>Filters>Mean. You have to enter the radius of the neighborhood in which the average for each pixel is computed. If the radius is 1 the neighborhood has a size of 3x3 pixels. Run the filter with different values for the radius and compare the results by zooming into the images. To run a mean filter with the radius 1 you can use Process>Smooth or Shift+S.

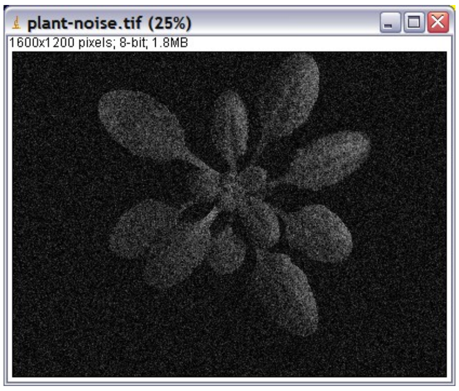


Illustration 54: An image containing noise.

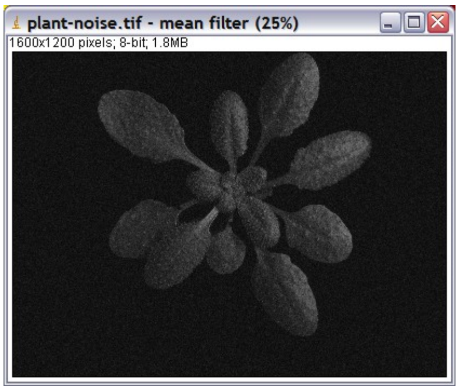

plant-noise.tif - mean filter (25%)
1600x1200 pixels; 8-bit; 1.8MB

Illustration 55: The image filtered with a mean filter of radius 1.

Open the image rectangle. You see a filled rectangle with intensity values 255 on a black background. What will be the intensity values of the edge and corner pixels after applying a mean filter with radius 1?

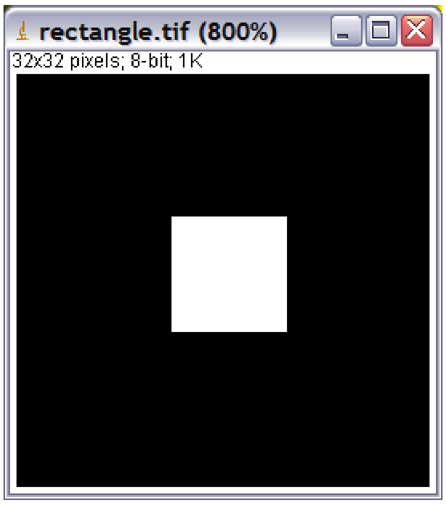


Illustration 56: A white rectangle.

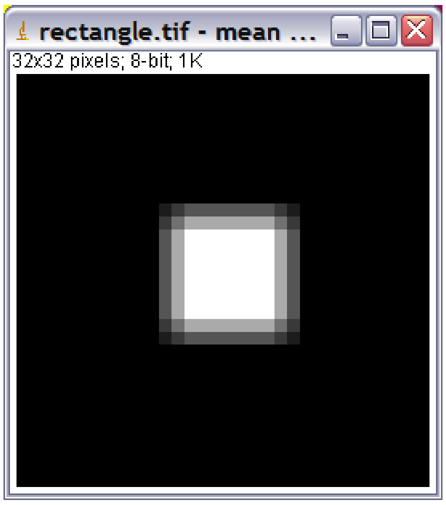


Illustration 57: The white rectangle after applying a mean filter of radius 1.

Compare the histograms of the original plant image and the plant image after a mean filter has been applied. Use a large radius, for example 15. Press h on the image or use Analyze>Histogram to display the histograms of the images.
Use the brightness and contrast adjuster to make the background as dark as possible and the plant as light as possible.

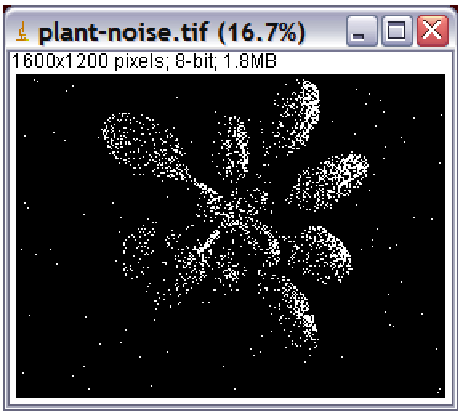


Illustration 58: Separation of object and background is impossible in the noisy image.

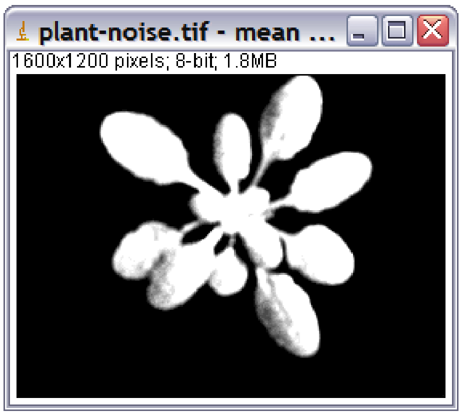
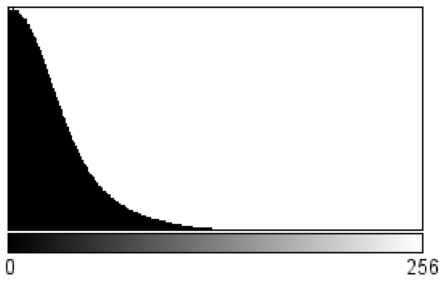
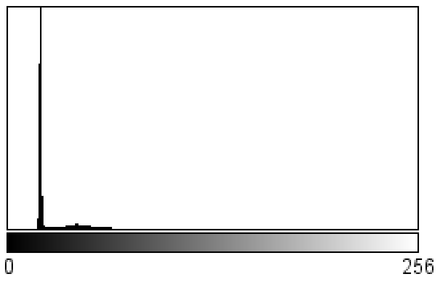


Illustration 59: The separation becomes possible after application of a mean filter.



```
Count: 1920000     Min: 0
Mean: 27.252       Max: 226
StdDev: 23.572     Mode: 2 (48226)
```

Illustration 60: The histogram of the noisy image.



```
Count: 1920000     Min: 15
Mean: 27.250       Max: 98
StdDev: 14.454     Mode: 20 (701069)
```

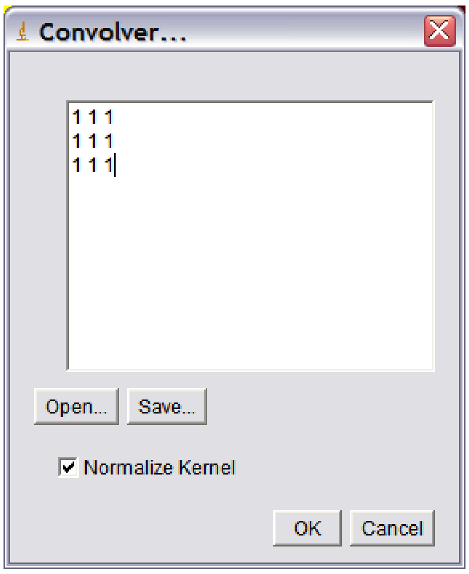
Illustration 61: The histogram after a mean filter of radius 15 has been applied.

A mean filter can be applied in the following way:

- For radius r create a matrix of size N = 2r+1 x 2r+1 filled with values 1/N. The matrix is called the kernel of the filter.
- Move the center of the kernel across the image. For each position:
  - Multiply each matrix element with the corresponding intensity value and calculate the sum of the results.
  - In the result image replace the intensity of the current pixel with the calculated result.

A generalization of this technique is called convolution. A convolution is an operation that calculates the overlap of two functions. In the general case the kernel and the image can have infinite size.
Instead of simply calculating the average, we can calculate a weighted average by using different values in the matrix. Convolution filters can not only be used for smoothing but also for other purposes, for example to enhance edges.
To apply a convolution filter go to Process>Filters>Convolve and enter the matrix in the dialog. To create a mean filter of radius 1 put in a 3x3 matrix, in which all elements are 1.

Normally we would have to use 1/9, but we can select the “normalize kernel” option and the computer will do this for us.



```
1 1 1
1 1 1
1 1 1
```

☑ Normalize Kernel

Illustration 62: Applying a mean filter by doing a convolution.

Apply the convolution filter. Compare the result with the result from the mean filter. To test whether the results are absolutely identical you can subtract one result image from the other. If the images were identical the result image must be all black. Use Process>ImageCalculator to subtract one image from the other. Apply the hilo lut to the result to see if intensities other than 0 are present.

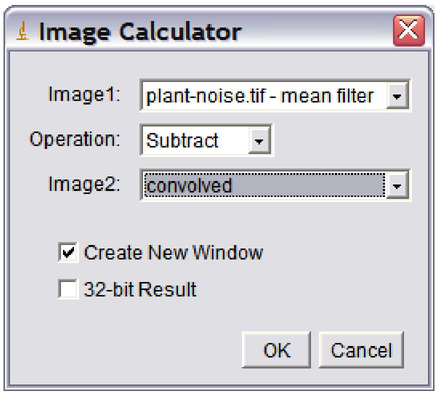


Image1: plant-noise.tif - mean filter
Operation: Subtract
Image2: convolved
☑ Create New Window
☐ 32-bit Result

Illustration 63: Subtracting one image from another image.

The smoothing reduces the deranging effect of noise but in the same time blurs the object, so that less details are visible.

4.6.1.2 Gaussian blur filter

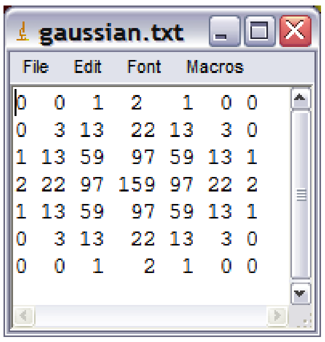
A Gaussian blur filter is similar to the mean filter, but instead of using uniform weights in the kernel, the weights are the values of a normal distribution, also called Gaussian distribution. A normal distribution is used to describe random processes. Throw a coin 100 times and count the times the head is up. Repeat this 1000 times and draw a histogram of the outcomes. Most of the time you'll get values around 50, resulting in high bars around 50 in the histogram. The case that in 100 throws only 1 head turns up will not occur very often. The same will be the case for 100 heads in 100 throws. If you connect the top points of the bars you get a curve that looks like a normal distribution and if instead of throwing the coin 1000 times you throw it an infinite number of times you'll get a normal distribution.
A two dimensional normal distribution has the form of a bell.

$$\varphi_{\mu,\sigma^2}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Illustration 64: A gaussian function.

In older versions of ImageJ the Gaussian Blur filter had the radius as a parameter. In newer versions (from 1.38q) the parameter is sigma, the standard deviation of the distribution. To get the same results the values in the old version have to be multiplied by 2.5.

Open the file *Gaussian.txt*. Here you see the values of a normal distribution. How does it look like as an image? Open the file as an image by using *File>Import>Text Image*. Apply the ice lut. Now go to *Analyze>Surface Plot* or use the *Interactive 3D surface plot* plugin (Plugins>3D) to get an impression of the 3D form. Because of the discrete nature, what you see looks less smooth then the real form. Besides the real curve never drops down to zero, but values will be very low outside the central area so that we can use a truncated Gaussian as an approximation.



```
0  0  1   2  1  0 0
0  3 13  22 13  3 0
1 13 59  97 59 13 1
2 22 97 159 97 22 2
1 13 59  97 59 13 1
0  3 13  22 13  3 0
0  0  1   2  1  0 0
```

Illustration 65: Convolution kernel for a (truncated) gaussian blur filter.

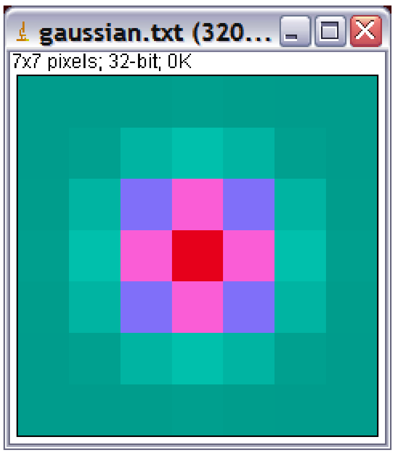


Illustration 66: The kernel as image.

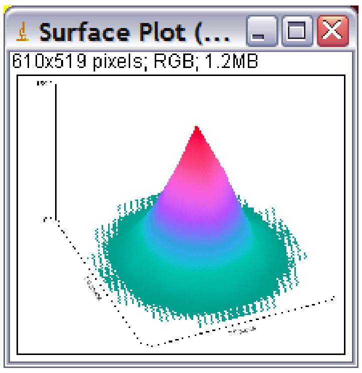


Illustration 67: The kernel as surface plot.

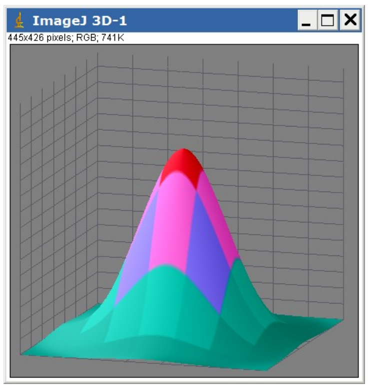


Illustration 68: The kernel as a surface plot created with the «interactive 3D surface plot» plugin.

Apply the Gaussian blur filter with sigma 1.2 from *Process>Filters>Gaussian blur* to the rectangle image and compare the result with the result of the mean filter with radius 3.

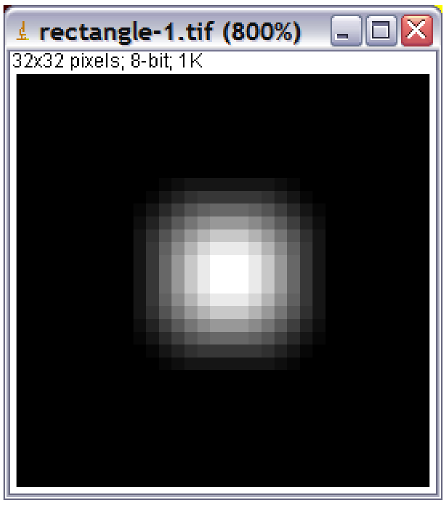


Illustration 69: Mean filter with sigma 1.2 applied to the rectangle image.

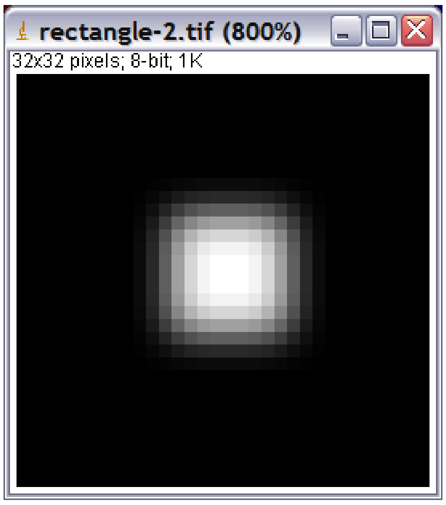


Illustration 70: Gaussian blur filter with radius 3 applied to the rectangle image.

Although both images are blurred, the result from the Gaussian filter looks more like the original form. The Gaussian filter while removing noise as well, keeps edges better. Apply a Gaussian blur filter with a sigma of 6 and a Mean filter of radius 15 to the image roots and compare the results.

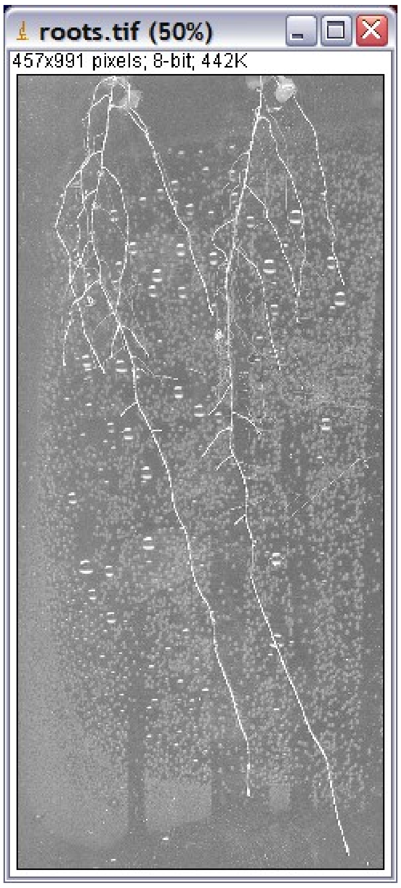


Illustration 71: The roots image.

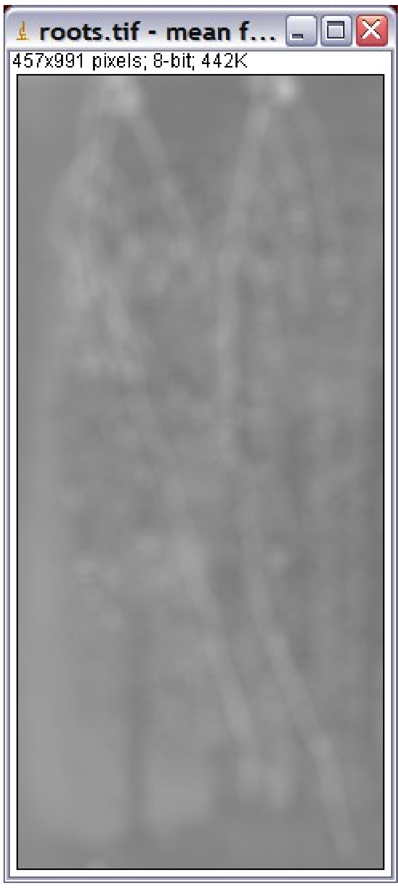


Illustration 72: A mean filter of radius 15 has been applied to the image.

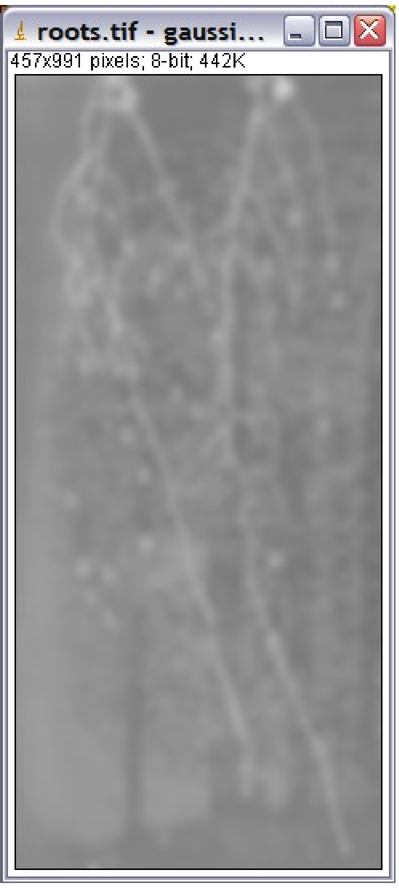


Illustration 73: A gaussian blur filter of radius 6 has been applied to the image.

Open the plant-noise image. Open the convolution tool and load the Gaussian.txt into it. Run the convolution filter on the image.

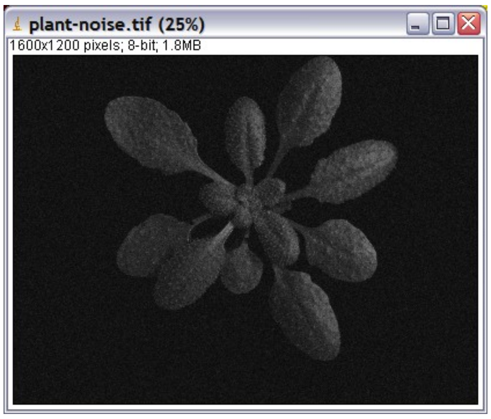
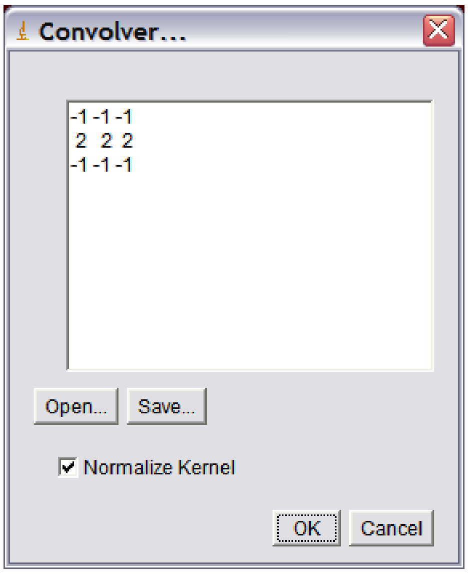


Illustration 74: The plant image after the convolution with the gaussian kernel has been applied.

4.6.1.3 Edge enhancing filter

Create a convolution filter of radius 1. Put -1 in the first row and in the last row, and 2 in the middle row. Apply it to the rectangle image and to the roots image. To see the effect of the filter, create an overlay of the original and the filtered image, by using *Image>Color>RGB Merge*. Put the filtered image in the red channel, the original image in the green channel and choose none for the blue channel.



```
-1 -1 -1
 2  2  2
-1 -1 -1
```

☑ Normalize Kernel

Illustration 75: The convolution kernal for an edge enhancing filter.

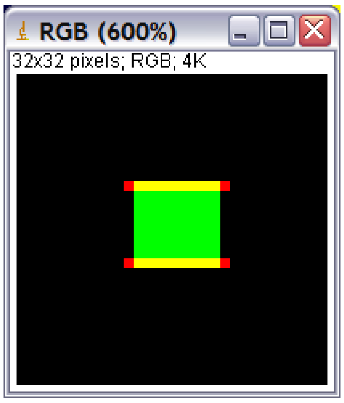


Illustration 76: RGB-merge of the original image and the result after application of the filter.

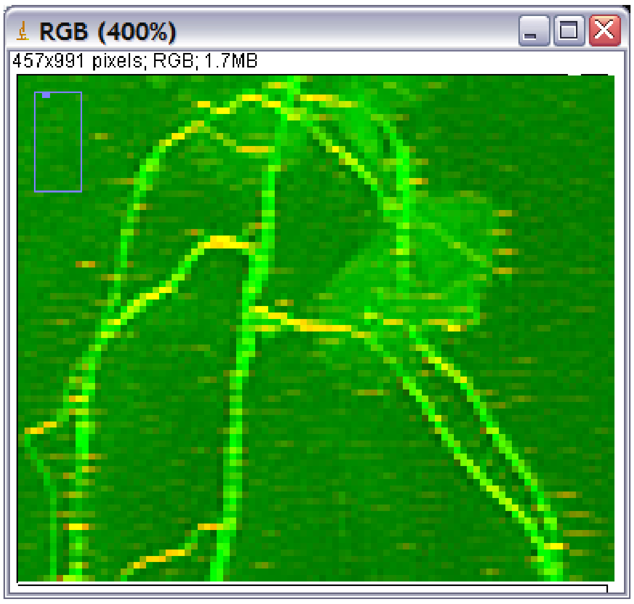


Illustration 77: Detail of the rgb-merge of the roots image and the result from the convolution that detects horizontal lines.

How will a filter that enhances vertical or 45 degree edges look like?

4.6.2 Rank Filter

Rank filters are another class of filters. Again we look at a neighborhood of radius r for each pixel. But this time the intensity values are sorted and the pixel is replaced with the value in a specific position.
If we use the middle position we get a median filter. If we use the first position we get a minimum filter and using the last position we get a maximum filter.
Open the image *plant-sp-noise.tif*. Zoom in to the image. You see that in this case the noise consists of the addition of white and black pixels to the image. This kind of noise is called salt-and-pepper noise.

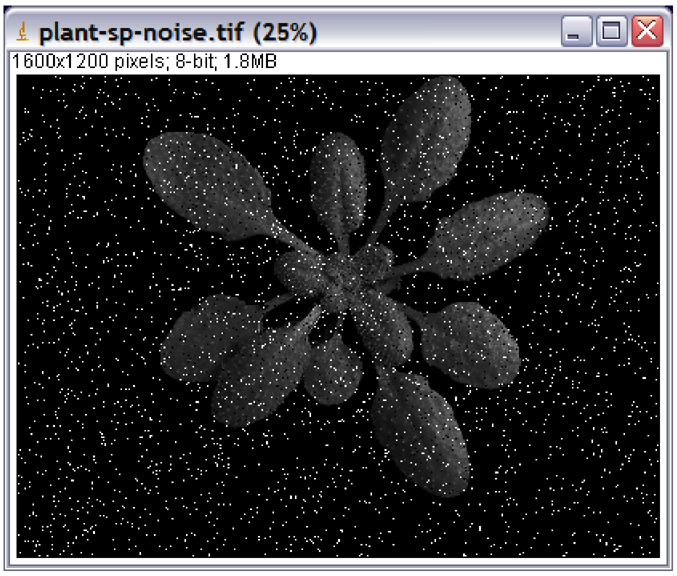


Illustration 78: The plant image with salt-and-pepper noise.

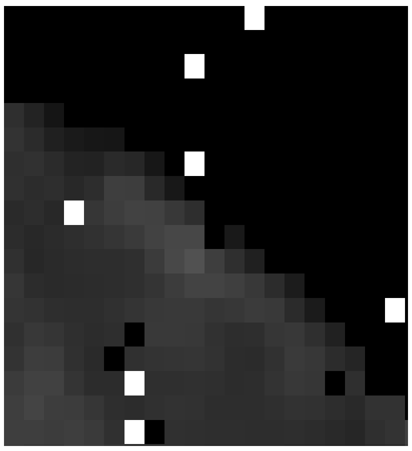


Illustration 79: Salt-and-pepper noise consisits of the addition of white and black pixels to the image.

Apply a median filter of radius 3 from Process>Filters>Median to the image. Compare the result with the result from the mean filter. The median filter is more effective in removing salt-and-pepper noise.

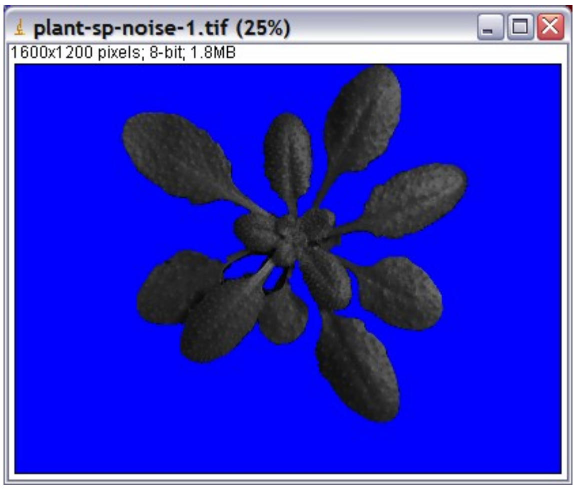


Illustration 80: A median filter effectively removes salt-and-pepper noise.

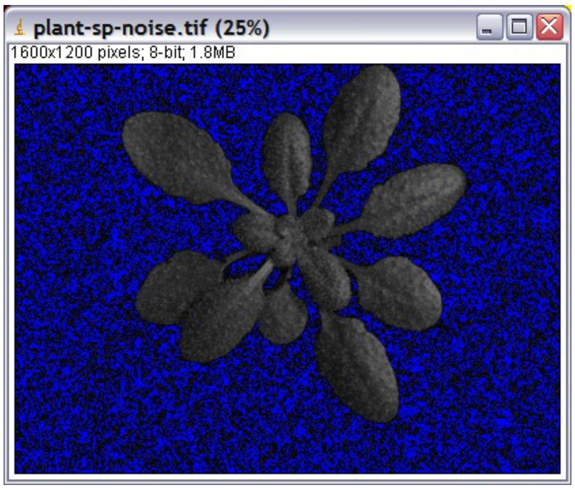


Illustration 81: A mean filter is not so effective on salt-and-pepper noise.

You can run a median filter with radius 1 using *Process>Noise>Despeckle*.
In *Process>Noise>Remove Outliers*, you find a selective median filter, that replaces a pixel by the median in the neighborhood if the value of the pixel differs more than a threshold value from the median. Use it to remove the dark and light components of the salt and pepper noise separately.

4.6.3 Filtering in the frequency domain

In its normal representation an image consists of an intensity value for each spatial coordinate. We call this the spatial domain. The same information can be represented in other ways. Some operations are easier to apply with a different representation.
One important representation is to transform the image into the (spatial) frequency domain. Here each value represents the frequency of change of the intensity value. Large structures have a low frequency, small repeating structures have a high frequency. The Fourier transform transforms the image from the spatial to the frequency domain. The transform is based on the fact that each signal can be presented as a sum of harmonic (sinoid) functions of different phase and amplitude. The result is an image of complex numbers that can be divided in a real part, the power spectrum and an imaginary part, the phase information. From the result the original image can be reconstructed without any loss by applying the inverse Fourier transform. Usually only the power spectrum is shown, since it contains most of the interesting information. However to reconstruct the original image, both, the power spectrum and the phase image are needed.
There exists an efficient way to compute a Fourier Transform. It is called the Fast Fourier Transform or FFT. We can use the Fourier Transform to filter our image in the frequency domain.
One reason why the frequency domain is interesting is because we will be able to recognize frequencies that appear as clusters visually. We can use this to filter out frequencies that correspond to objects or patterns of different size in the image.
Low frequencies will be displayed in the middle of the power spectrum and the higher the frequencies are, the bigger is their distance to the center. If we filter out the low frequencies in

the middle, we apply a high pass filter. This can be used as an edge enhancing filter since the edges information is in the high frequencies.
If we filter out the high frequencies, we apply a low pass filter. This has a smoothing effect on the image, since the fine details have a high frequency.
Another reason why the Fourier Transform is important is because we can calculate a convolution more efficiently in the Fourier Domain. The Convolution Theorem states that a convolution in the spatial domain is equivalent to a multiplication in the frequency domain.
Open the image *Pierre.tif*. Apply the Fourier transform from *Process>FFT>FFT*.

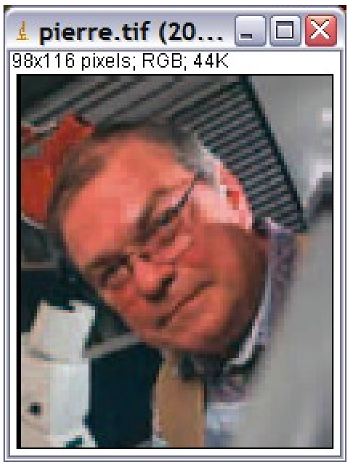


Illustration 82: An image.

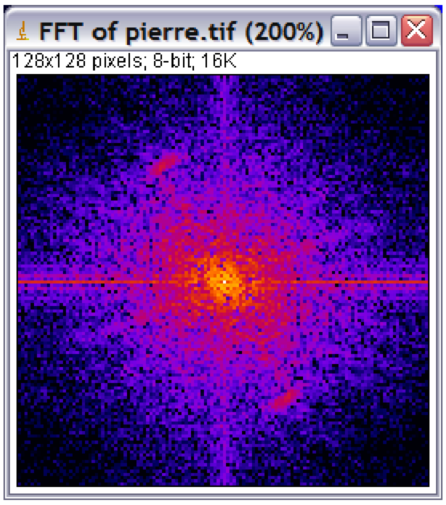


Illustration 83: The power spectrum of the image.

You can use the lut tool to make visible different clusters of frequencies. Do you see the two clusters in the upper left and lower right quadrant. Draw a selection around the first. Then add a selection around the second by pressing shift when you start the selection. You can release the shift key once you started the second selection. Now run the fill command from *Edit>Fill* or press *CTRL+f*. We have set the intensities in the selection to zero and in this way suppressed the corresponding frequencies. Apply the inverse Fourier transform from *Process>FFT>Inverse FFT*. At first look the image doesn't seem to have changed. Zoom in on the image and on the result image behind the ear. You see that the lines of the window blinds have disappeared in the filtered image, since we suppressed the corresponding frequencies.

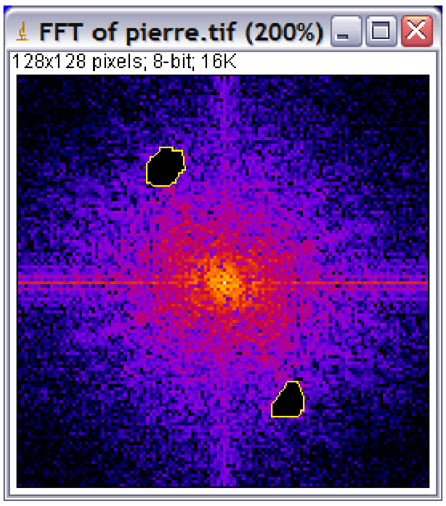


Illustration 84: Filling parts of the power spectrum image with the background color filters out the corresponding frequencies.

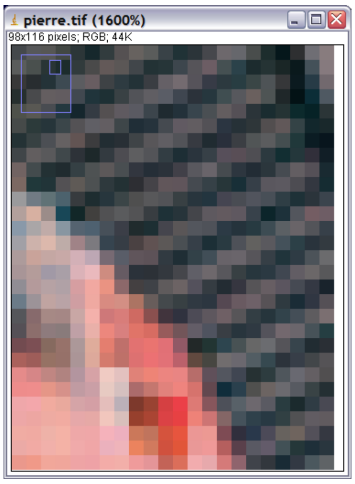


Illustration 85: Detail of the original image.

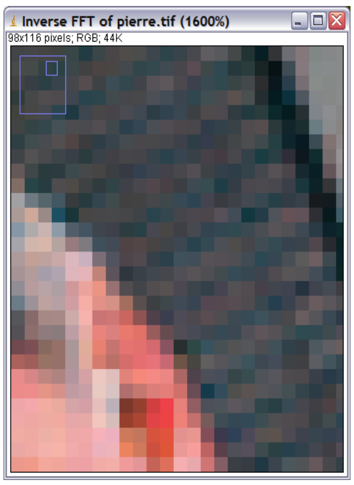


Illustration 86: Detail of the filtered image.

Subtract the filtered image from the input image to see what exactly has been filtered out. Instead of building the difference you can as well go back to the power spectrum and fill the inverse of the mask.

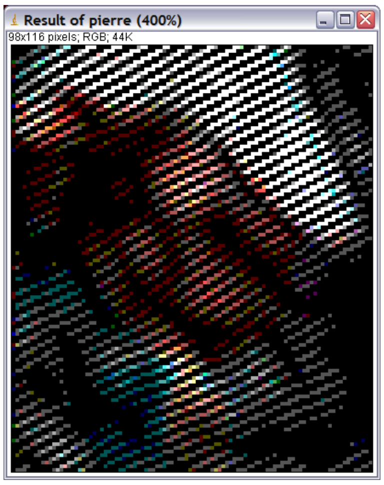


Illustration 87: The difference of the original image and the filtered image.

Use *Process>FFT>Redisplay Powerspectrum* to redisplay the original power spectrum. Try to smooth the plant-noise image. Which frequencies do you have to filter out?

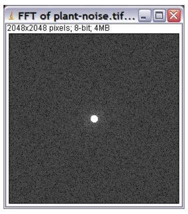


Illustration 88: Using clear instead of fill lets the corresponding frequencies pass.

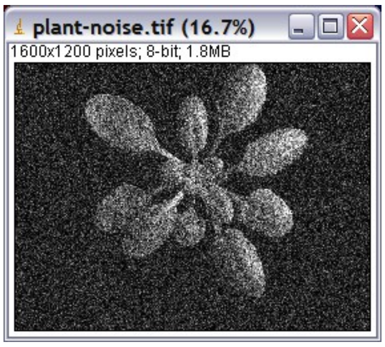


Illustration 89: The original image.

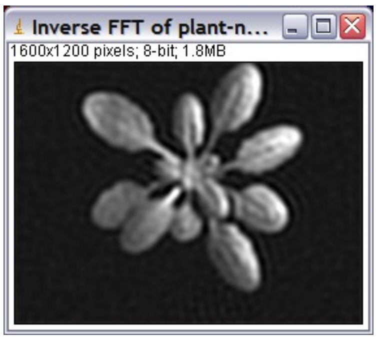


Illustration 90: The image after filtering in the frequency domain.

Try to enhance the edges in the roots image. Always build the difference of the original image and the filtered image to see what exactly has been filtered out.

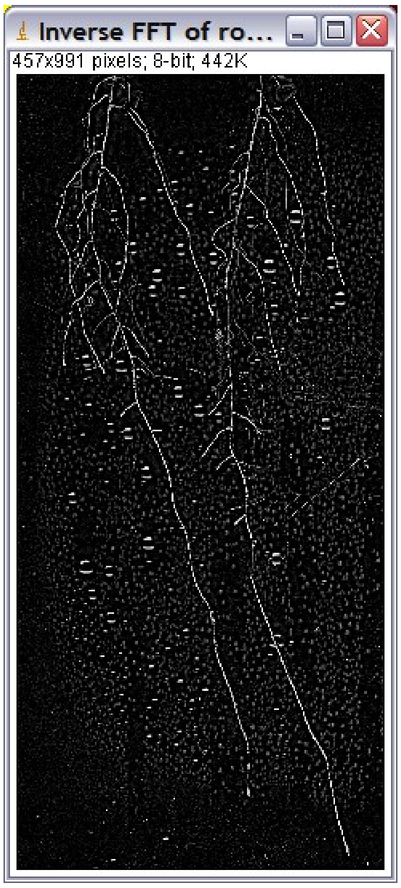


Illustration 91: The roots image after a high pass filter has been applied.

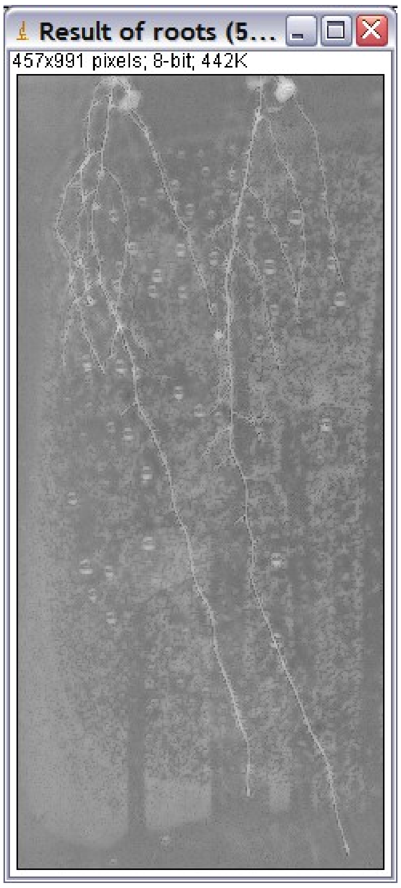


Illustration 92: The difference of the original and the filtered image.

If we set everything, except for a ring, to zero in the power spectrum, we will filter out all frequencies higher then a certain value and all frequencies lower then a certain value. This is called a band-pass filter, since the only frequencies that pass are those within the ring.
Try it for example with the image pierre.tif . Draw a circular selection and then remove a second circle from within the selection by pressing the alt key while making the second selection. Use the I from the toolbox to invert the selection and apply fill from the context menu.

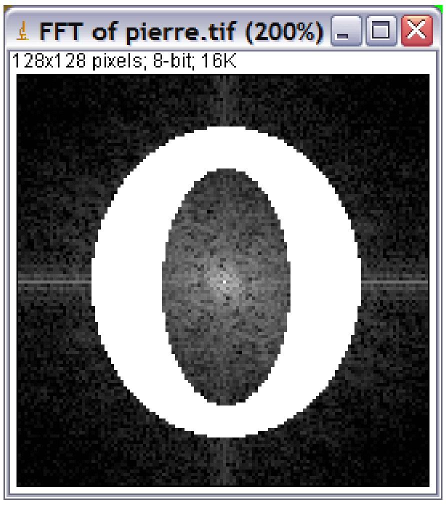


Illustration 93: Defining a bandpass filter.

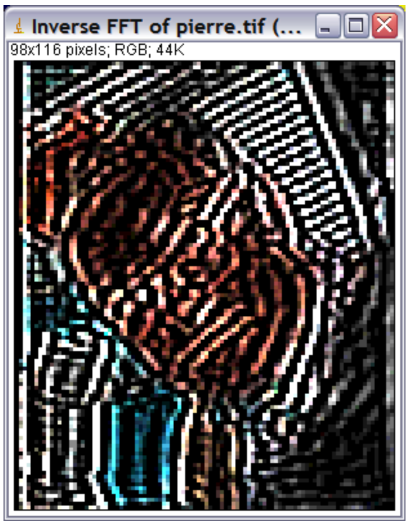


Illustration 94: The bandpass filter applied to an image.

4.7 Background subtraction

Open the image *nuclei-rhod.tif*. Apply the *hilo lut*. As you see no pixels become blue, meaning that the background is not zero. Since this image is from fluorescent microscopy, normally the background should be black, in reality however there is some intensity in the background.
In this example the background is low and, except for the noise, homogeneous. A high background can make it harder to clearly distinguish objects. Furthermore if we want to compare intensities between images with different background levels we have to compensate for the background.
Use the B&C tool to clearly distinguish a background region. Draw a line selection on the background and create a profile plot using Analyze>Plot profile or select the image and press k. The profile plot shows the intensity values along the line selection. You see that the values vary somewhere between 1 and 9 due to noise and that the average is somewhere around 3. Press the measure button on the toolbox or select the image and press ctrl+m to get the exact mean value.

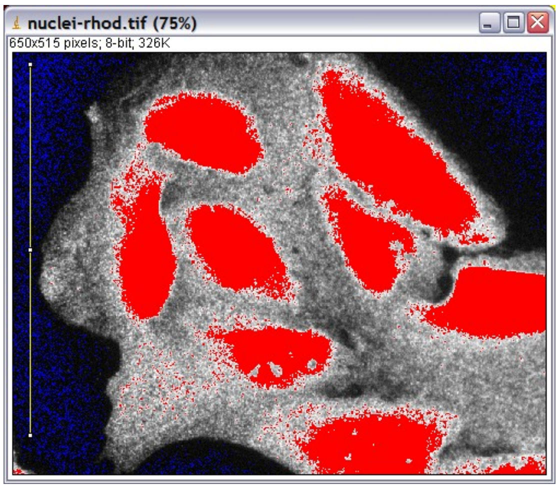


Illustration 95: The background should be black, but contains some intensity in reality.

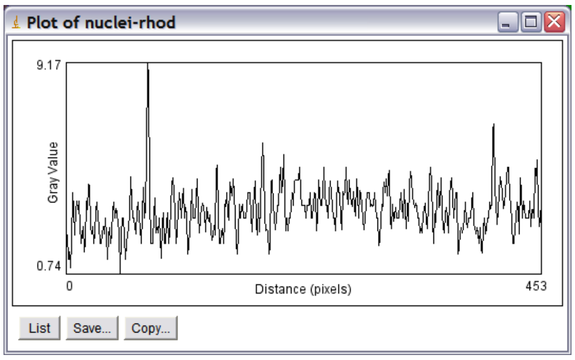


Illustration 96: The line plot displays the intensities along the selected line.

Go to Process>Math>subtract and subtract the mean background value from the image. Use the R button from the toolbox to get back the line selection and create the profile plot again. This time the values go down to zero. The command *Plugins>ROI>BG Subtraction from Roi* subtracts the next integer value, above the mean intensity within the selection, from the image.

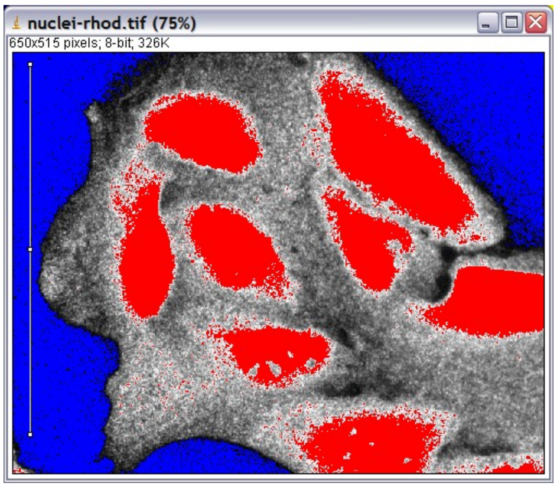


Illustration 97: The image after the value 3 has been subtracted from each pixel value..

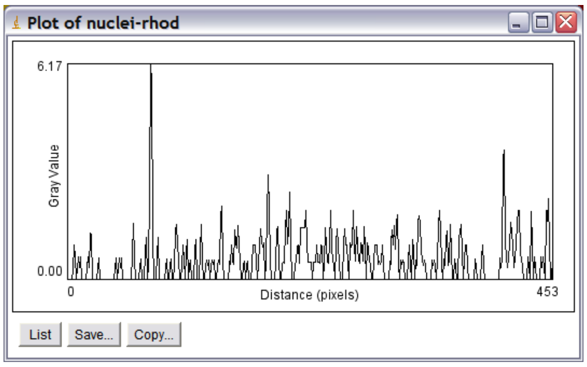


Illustration 98: The line plot goes down to zero now.

Another way to get rid of the background is to divide the image by the average background value. Most background values will be 1 afterwards and you can subtract 1 from the image to set them to zero. Try it with *Process>Math>divide*.

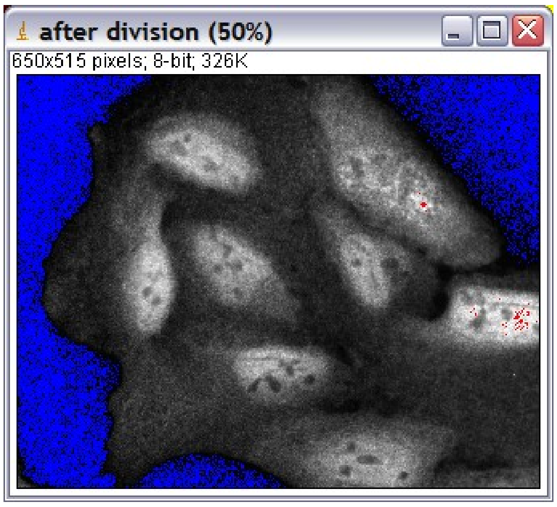


Illustration 99: The image after it has been divided by 3.3 and 1 has been subtracted.

There is an automatic operation that tries to find and subtract the background. Open the visual scripting plugin from *Plugins>Montpellier RIO Imaging>MRI VisualScripting*. Go to *Operations>All* and search the Find and Subtract Background operation. Drag it from the list and press the Options button (O). Set the number of iterations to 1. Run the operation by pressing the button in the middle. The operation searches for the highest intensity value around the intensity minima and subtracts it from the image.

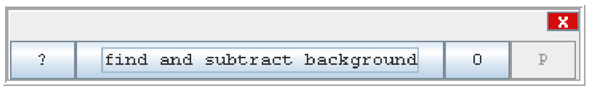
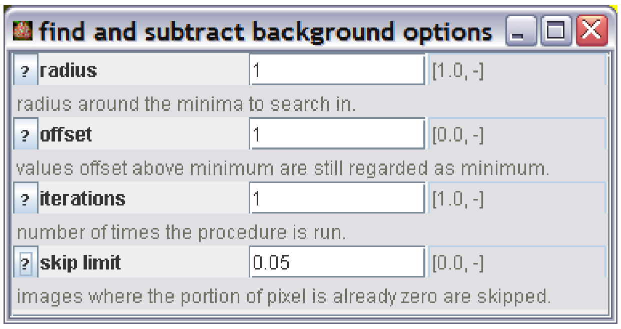


Illustration 100: The find and subtract background operation.



find and subtract background options

? radius	1	[1.0, -]
radius around the minima to search in.		
? offset	1	[0.0, -]
values offset above minimum are still regarded as minimum.		
? iterations	1	[1.0, -]
number of times the procedure is run.		
? skip limit	0.05	[0.0, -]
images where the portion of pixel is already zero are skipped.		

Illustration 101: The options dialog of the operation.

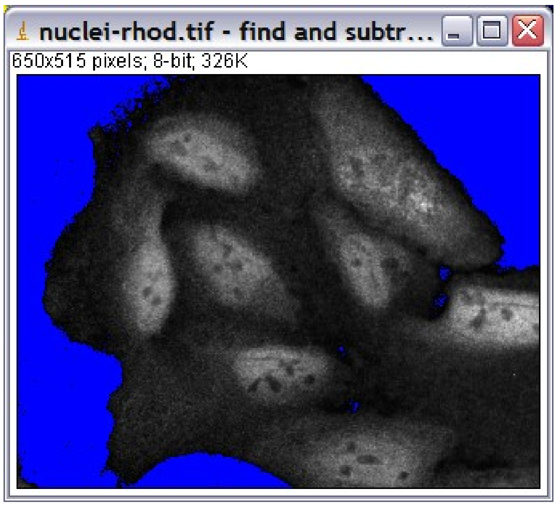

nuclei-rhod.tif - find and subtr...
650x515 pixels; 8-bit; 326K

Illustration 102: The image after the find and subtract background operation has been applied.

4.7.1 Inhomogeneous background

Open the image *root2.tif*. Draw a vertical line on the background again and create the profile plot. As you see this time the background baseline is not constant. The image is darker in the top area and lighter in the bottom area. Measure the mean and subtract it from the image. Create the profile plot again. You see that, although the background value is lower now, the gradient is still present.

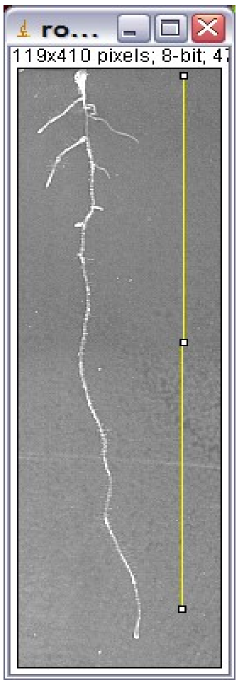


Illustration 103: An image with an inhomogeneous background.

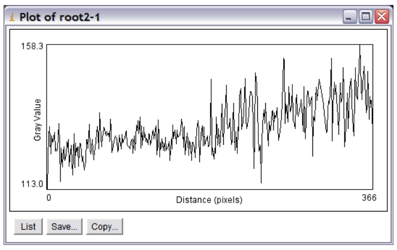


Illustration 104: The line plot reveals a gradient from the top to the bottom of the image.

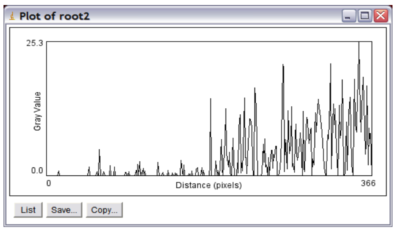
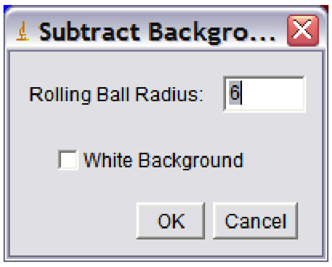


Illustration 105: After subtraction of the average background intensity the gradient is still present.

Apply the operation *Process>Subtract Background* to the image and create the profile plot. The gradient has disappeared. The command uses a rolling ball algorithm that, roughly described, moves a sphere along the image and considers intensity values outside the radius to be background. In this way the local information is taken into account.



Subtract Backgro...

Rolling Ball Radius: 6

White Background

Illustration 106: The options dialog of the subtract background command.

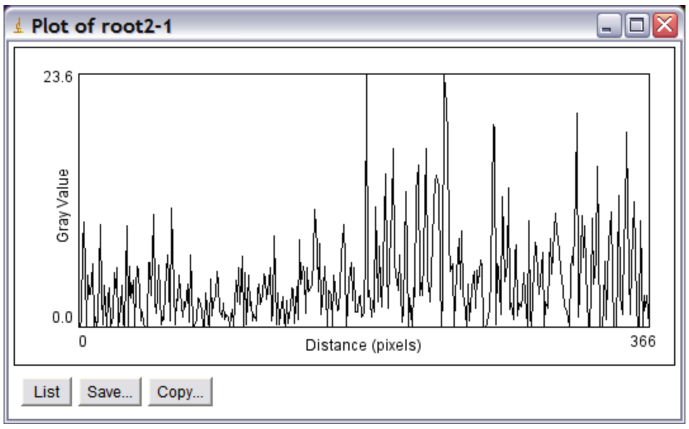
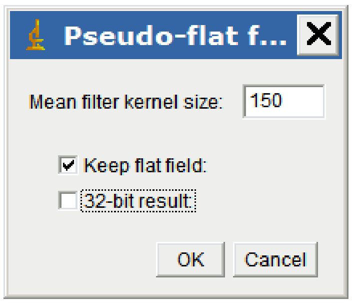


Illustration 107: The subtract background command removes the gradient using a rolling ball algorithm.

One very elegant way to remove background is to make an image of the empty scene along with the image of the specimen. One can than divide the specimen image by the background image and multiply the result with the mean intensity in the background image . This is often possible in microscopy and the technique is called flat field correction. If it is not possible to take the “empty” image we can simulate the process by applying a very large filter that will remove the object from the image. We then have to divide the result from the original image and multiply with the average intensity of the filtered image. This can be done with *Process>Filters>Pseudo Flat Field*. Select the keep flat field box to see the generated background image. Look at the profile plot from the result. Again the gradient has disappeared. The Pseudo Flat Field correction applies a mean filter to the image. You can do the same thing using other filters. Try a Gaussian blur filter for example.


Pseudo-flat f...

Mean filter kernel size: 150

[✓] Keep flat field:

[ ] 32-bit result

Illustration 108: The pseudo flatdield correction options dialog.

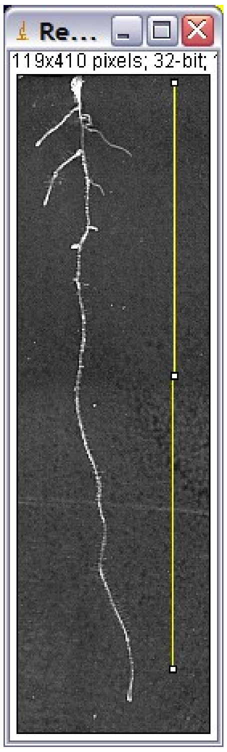


Illustration 109: The image after pseudo flatfield correction.

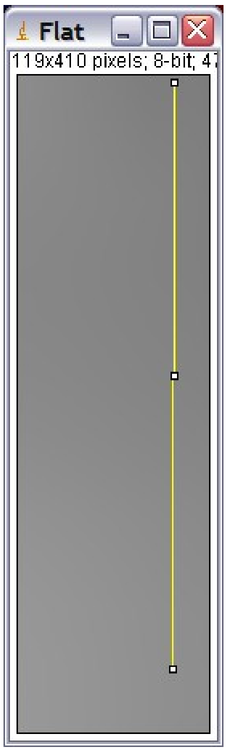


Illustration 110: The generated background that is subtracted from the image.

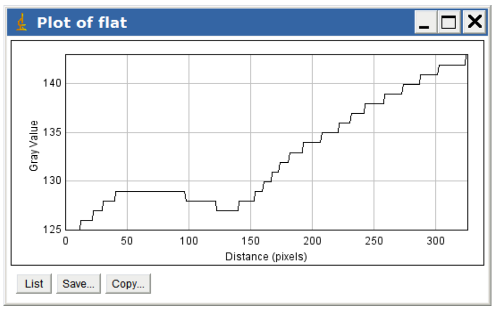


Illustration 111: The line plot of the generated background.

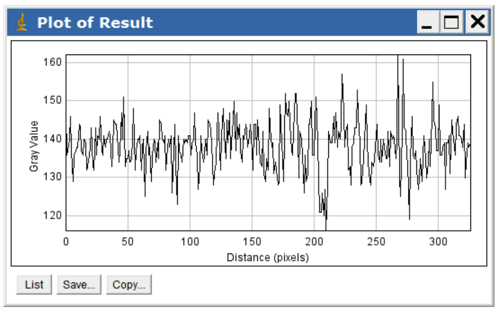


Illustration 112: Line plot after pseudo flatfield correction. The gradient has disappeared.

4.8 Increasing the apparent sharpness

Open the image hst2.tif from the folder “cells in structures”. Zoom in 2 or 3 times. The image appears somewhat blurred or unsharp. Duplicate the image with the duplicate button from the tool box and apply a Gaussian blur with sigma 0.8. Multiply the result with the factor 0.6 and subtract the result from the original image.
This technique is called unsharp masking. You can use *Process>Filters>Unsharp* Mask to do it in one step.

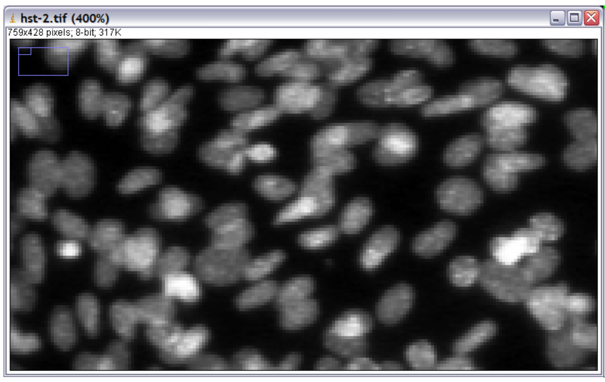


Illustration 113: The image appears somewhat unsharp or blurred.

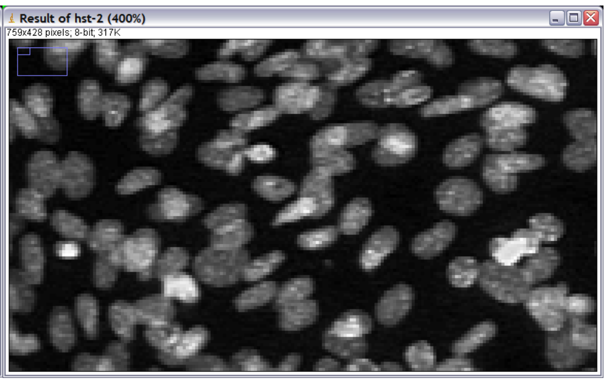


Illustration 114: After applying the unsharp mask filter the image appears less blurred.

5. Segmentation

Segmentation is the process of separating the objects from the background and from one another. The simplest way of segmentation is based on the difference in the intensities of objects and background. One looks for a minimum and maximum intensity value, so that all pixels belonging to the objects one is interested in have intensities between min and max and all other pixels have intensity values below min or above max.
The result of a segmentation is often represented as a binary mask. A binary mask is an image that contains only two values. The first value signifies that the pixel belongs to the background and the second signifies that a pixel belongs to the object. In ImageJ the values 0 and 255 are used. Once we created a masked, we can combine it with the original image, for example to suppress everything but the objects we are interested in.
Another way of presenting the result of a segmentation is a selection. The selection represents the borders of the objects we are interested in. A selection itself is not an image. ImageJ stores the selection together with the image. A selection is called a region of interest or ROI in ImageJ. We can turn a mask into a selection and vice versa and use the representation the most practical for a given task. A selection can be created from a mask with the help of the magic wand or tracing tool. Given a selection, a mask can be created by using fill on the mask, then inverting the mask and using clear. The command *Edit>Selection>Create mask* can be used to create a mask from a selection in one step. The command *Edit>Selection>Create Selection* can be used to create a selection from a mask or from an image on which the lower and upper threshold are set in one step.

5.1 Manual threshold selection

The threshold value(s) can be either chosen manually or automatically by an algorithm. The most simple automatic thresholding algorithms try to find a threshold that divides the histogram in a meaningful way.
Open the image *plant.tif*. Open the threshold adjuster from *Image>Adjust>Threshold* or click shift+t on the image. You can choose between three modes in the threshold tool. The first displays selected pixel in red and pixels not selected with their original intensities. The second displays the mask that will be the result after applying the threshold, i.e. background in white and objects in black. In the third mode, selected pixels are displayed with their intensities, pixels below the lower threshold are displayed in blue and pixels above the upper threshold are displayed in green.

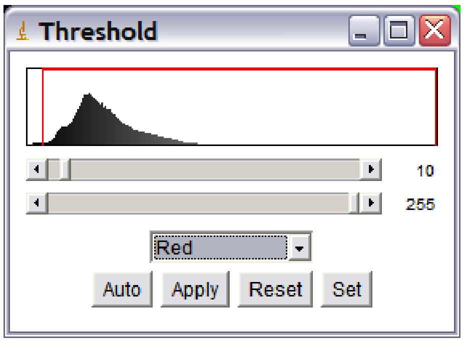


Illustration 115: The threshold adjuster.

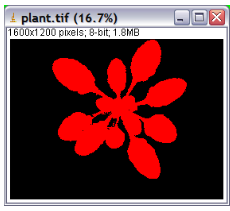


Illustration 116: Object red, background black.

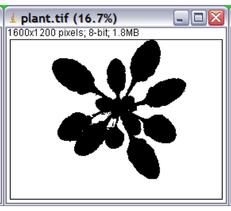


Illustration 117: Object black, background white.

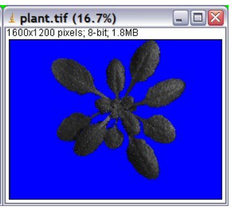


Illustration 118: Object graylevels, background blue (below first threshold) and green (above second threshold) .

Adjust the threshold to select the plant. The wand tool works together with the threshold tool. So if you want to select the plant, there is no need to create the mask. Use the wand tool to select the plant. Use the alt key to keep out the two holes between the leaves. You can add to a selection by using shift and subtract from a selection by using alt. Alternatively you can use *Edit>Selection>Create Selection* after you changed the lower and upper threshold with the threshold adjuster. In contrast to the create selection command, using the wand tool allows to select one separated object in an image containing multiple objects.
If you press the measure button on the toolbox or the m key on the image, you get a result table with measurements of the selection.

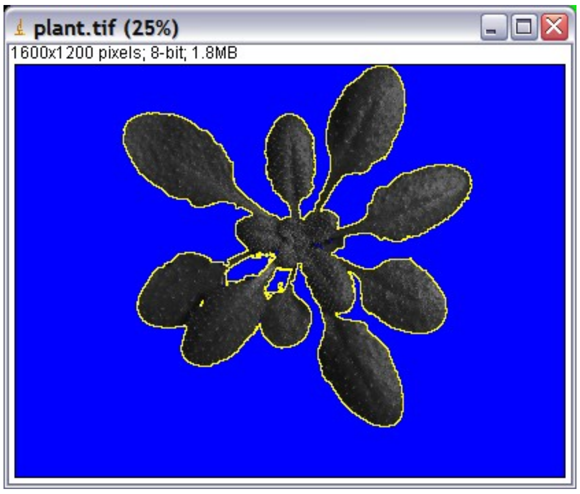
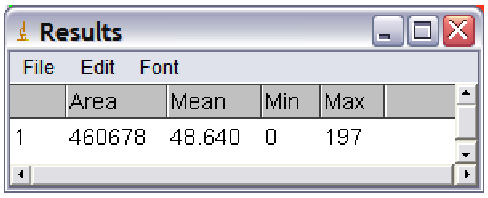


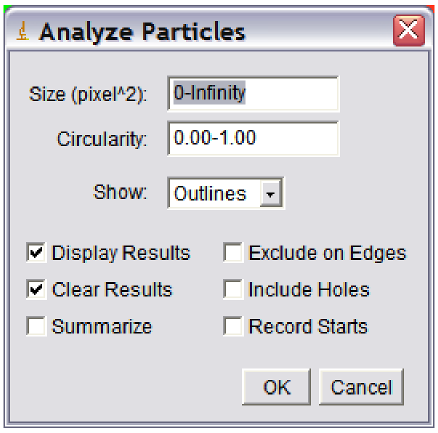
Illustration 119: Selection created with the threshold adjuster and the wand tool.



	Area	Mean	Min	Max
1	460678	48.640	0	197

Illustration 120: Results table with the measurements from the selection.

Remove the selection by clicking on the N button in the toolbox. Run the Particle Analyzer to measure all objects with intensities between the selected thresholds. Use the A*pply* button if you want to create a mask.



Size (pixel^2): 0-Infinity
Circularity: 0.00-1.00
Show: Outliness

[✓] Display Results
[✓] Clear Results
[ ] Summarize
[ ] Exclude on Edges
[ ] Include Holes
[ ] Record Starts

Illustration 121: The particle analyzer.

You can filter out objects by their size and circularity and you can create a mask image or an outline image containing the contours of the measured objects. In the later image the objects will be numbered, as well. You can choose to exclude objects touching the image borders and to include or not include holes in the objects in the mask.

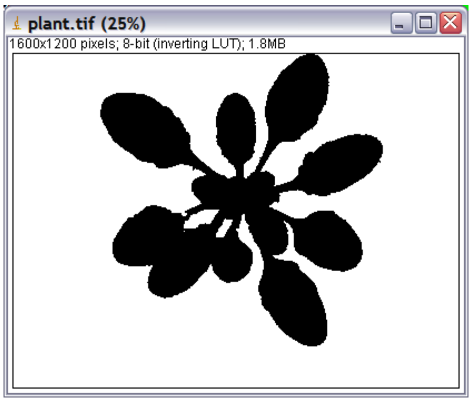


Illustration 122: A mask for the plant image.

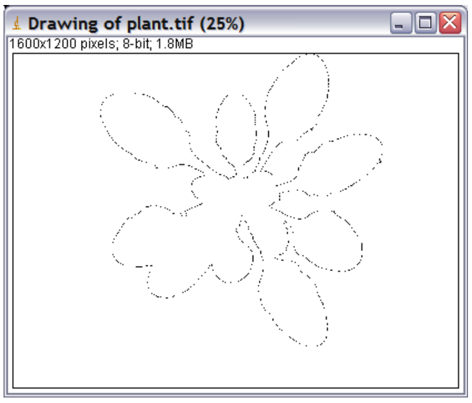
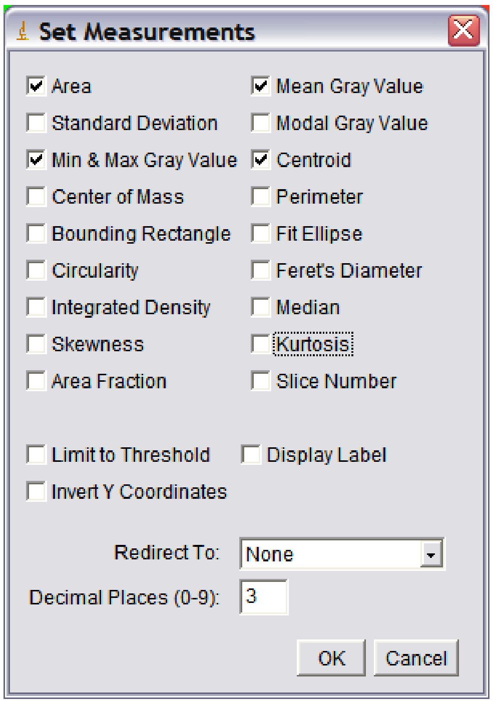


Illustration 123: An outline of the plant created with the particle analyzer.

You can configure the values that will be measured by the Particle Analyzer under *Analyze>Set Measurements* or under *Options>Measurements* on the toolbox. Here you can as well configure a redirect image. In this way you can use a mask but the intensities will be measured in the redirect image, for example the original image the mask was created from. Use the Particle Analyzer to measure the plant with the intensity values from the original image and display the outline result.



Set Measurements

- [✓] Area
- [✓] Mean Gray Value
- [ ] Standard Deviation
- [ ] Modal Gray Value
- [✓] Min & Max Gray Value
- [✓] Centroid
- [ ] Center of Mass
- [ ] Perimeter
- [ ] Bounding Rectangle
- [ ] Fit Ellipse
- [ ] Circularity
- [ ] Feret's Diameter
- [ ] Integrated Density
- [ ] Median
- [ ] Skewness
- [ ] Kurtosis
- [ ] Area Fraction
- [ ] Slice Number
- [ ] Limit to Threshold
- [ ] Display Label
- [ ] Invert Y Coordinates

Redirect To: None

Decimal Places (0-9): 3

Illustration 124: The set measurements dialog. The settings are used by the measure command and by the particle analyzer.

Another way to use a mask to measure in another image, is by using the wand tool on the mask and then transferring the mask to the other image by activating the image and pressing R on the toolbox or shift+e on the image. You can then use the measure command to measure the content of the selection.
Open the *plant-noise.tif* image and try to segment it with the help of the threshold adjuster. You will see that the noise prevents you from making a good segmentation. Try one of the noise suppression techniques we discussed and try the segmentation again.

5.2 Measurements and the ROI-Manager

Open the image *A4 dapi1.tif.* Set a threshold that separates the nuclei from the background. Run the *Particle Analyzer.* Select the *Add to Manager* option. For each particle detected by the Particle Analyzer, a roi is added to the ROI-Manager. The *Show All* button shows or hides all rois in the manager on the active image.
If you click on one roi in the list, it will be set to the active image. Clicking on a label of a roi in the image, selects the roi in the list. Now deselect all rois and press the measure button. You get a results table with the measurements of all rois. If you select a number of rois using shift, only the selected rois will be measured. In general the commands in the roi manager are either applied to the selected rois or to all rois if there is no selection.

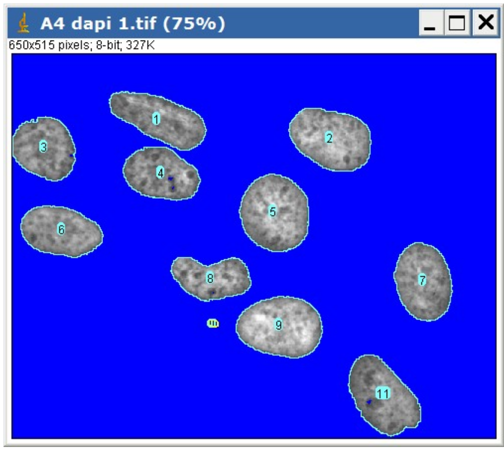


Illustration 125: The roi manager displays the rois of all particles.

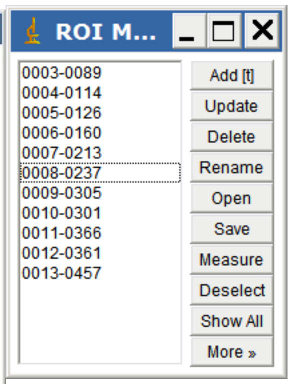
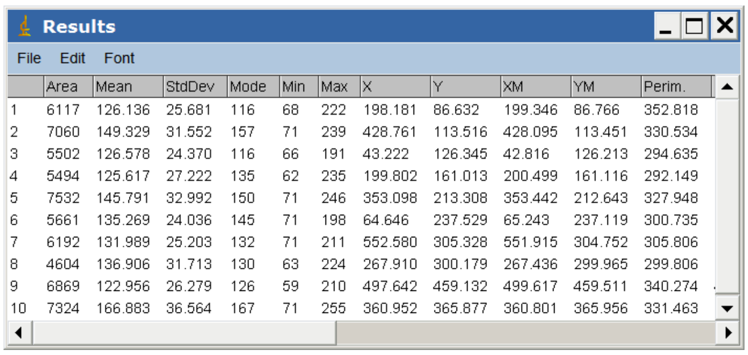


Illustration 126: The rois of the particles have been added to the roi manager.



	Area	Mean	StdDev	Mode	Min	Max	X	Y	XM	YM	Perim.
1	6117	126.136	25.681	116	68	222	198.181	86.632	199.346	86.766	352.818
2	7060	149.329	31.552	157	71	239	428.761	113.516	428.095	113.451	330.534
3	5502	126.578	24.370	116	66	191	43.222	126.345	42.816	126.213	294.635
4	5494	125.617	27.222	135	62	235	199.802	161.013	200.499	161.116	292.149
5	7532	145.791	32.992	150	71	246	353.098	213.308	353.442	212.643	327.948
6	5661	135.269	24.036	145	71	198	64.646	237.529	65.243	237.119	300.735
7	6192	131.989	25.203	132	71	211	552.580	305.328	551.915	304.752	305.806
8	4604	136.906	31.713	130	63	224	267.910	300.179	267.436	299.965	299.806
9	6869	122.956	26.279	126	59	210	497.642	459.132	499.617	459.511	340.274
10	7324	166.883	36.564	167	71	255	360.952	365.877	360.801	365.956	331.463

Illustration 127: The measurements of the rois in the roi manager.

Remove particle 10 and 9. Use the wand tool to select particle 10 again and add it again to the roi manager.
Save all rois, then close and re-open the roi-manager and open the saved rois again.
Compare the measured values of different particles. What is the meaning of the measured values?

Area:
The surface of the roi. It is measured in square-pixel if no spacial scale is set.

Mean:
The average gray value within the roi.

StdDev:
The standard deviation of the mean gray value within the roi.

Mode:
The most frequently occurring gray value within the roi. This corresponds to the highest peak in the histogram of the roi. If you display the histogram the mode will be shown together with the number of occurrences.

Min:
The minimal gray value within the roi.

Max:
The maximum gray value within the roi.

X:
The x-coordinate of the centroid. This is the average of the x-coordinates of the pixels in the roi.

Y:
The y-coordinate of the centroid. This is the average of the y-coordinates of the pixels in the roi.

XM:
The x-coordinate of the center of mass. This is the brightness weighted average of the x-coordinates of the pixels in the roi.

YM:
The x-coordinate of the center of mass. This is the brightness weighted average of the y-coordinates of the pixels in the roi.

Perim.:
The length of the outside boundary of the roi.

BX:
The x-coordinate of the upper-left corner of the bounding box of the roi. The bounding box is the smallest rectangle, that entirely contains the roi.

BY:
The y-coordinate of the upper-left corner of the bounding box of the roi.

Width:
The width of the bounding box of the roi.

Height:
The height of the bounding box of the roi.

Major:
The length of the major axis of the best fitting ellipse. The ellipse has the same area, orientation and centroid as the original selection.

Minor:
The length of the minor axis of the best fitting ellipse.

Angle:
The angle of the major axis of the best fitting ellipse against the x-axis of the image.

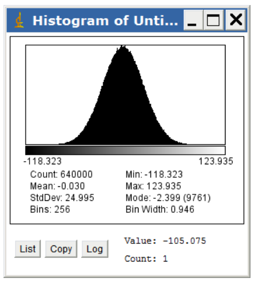
Circularity:
4π(area/perimeter^2). A value of 1 indicates a perfect circle. As the value approaches 0, it indicates an increasingly elongated polygon. Values may not be valid for very small particles. The perimeter of a circle is p = 2πr and the area is a = πr².

Feret:
The longest distance between two points on the boundary of the roi.

IntDen:
The integrated density is the sum of the gray-values of all pixels within the roi.

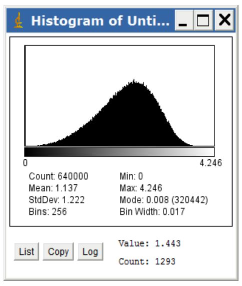
Median:
The median gray value of the pixels within the roi, i.e. The gray value that lies in the middle or the value between the two values in the middle, when all gray-values are sorted by their numerical value.

Skewness:
A measure of the asymmetry of the distribution of the gray values around the mean within the roi.



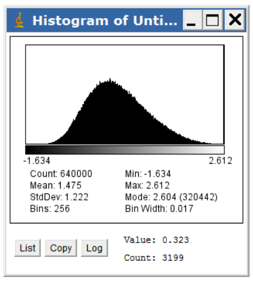
```
Count: 640000      Min: -118.323
Mean: -0.030       Max: 123.935
StdDev: 24.995     Mode: -2.399 (9761)
Bins: 256          Bin Width: 0.946
Value: -105.075
Count: 1
```

Illustration 128: The measured skewness is 0.001



```
Count: 640000      Min: 0
Mean: 1.137        Max: 4.246
StdDev: 1.222      Mode: 0.008 (320442)
Bins: 256          Bin Width: 0.017
Value: 1.443
Count: 1293
```

Illustration 129: The measured skewness is 0.345



```
Count: 640000      Min: -1.634
Mean: 1.475        Max: 2.612
StdDev: 1.222      Mode: 2.604 (320442)
Bins: 256          Bin Width: 0.017
Value: 0.323
Count: 3199
```

Illustration 130: The measured skewness is -0.345

Kurtosis:
A measure of the "peakedness" of the distribution of the gray values around the mean within the roi. Higher kurtosis means more of the variance is due to infrequent extreme deviations, as opposed to frequent modestly-sized deviations.

5.2 Computed thresholds

Open the *plant.tif* image again. We will now have a look at the automatic threshold operations. Open the MRI Visual Scripting Tool. Go to *Operations>basic>segmentation*. Try the auto-threshold, the entropy-threshold, the mean-threshold and the otsu-threshold. Which one gives the best result? Try them on the *root2.tif* image. Which one gives the best result for this image?

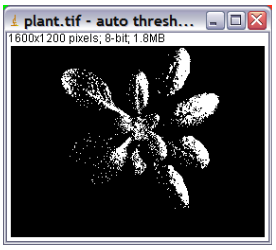


Illustration 131: Result of the auto-threshold.

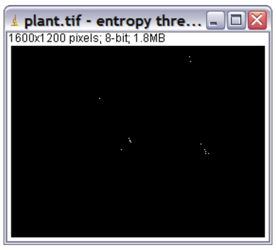


Illustration 132: Result of the entropy-threshold.

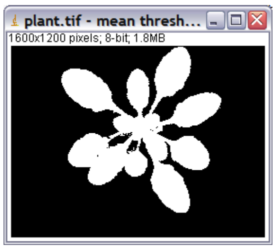


Illustration 133: Result of the mean-threshold.

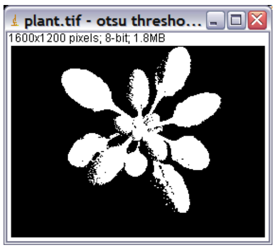


Illustration 134: Result of the Otsu-threshold.

The auto-threshold separates the histogram in a way that the threshold value equals half of the sum, of the average background and the average object intensity. In other words it is trying to find a threshold in a way that half of the average intensity belongs to background and half of it to objects.
The entropy threshold searches a threshold value for which the inter class entropy is maximal.
The Otsu threshold uses a threshold value for which the inter class variance is maximal.
The mean threshold uses the mean intensity of the image as threshold value.

5.3 Segmentation of more then 2 object classes

We can have more then two classes for background and objects. We could for example have background, dark objects and light objects.
Go to *Plugins>Segmentation>k-means-clustering* and apply it to the plant image, using 4 classes. The result is an image with the intensities 1, 2, 3 and 4. Pixels with a similar intensity are marked with the same number. We could now multiply the image with 60 and then use the threshold adjuster to create a mask from each of the numbers. Two binary masks can be combined by using Process>Image Calculator>And.

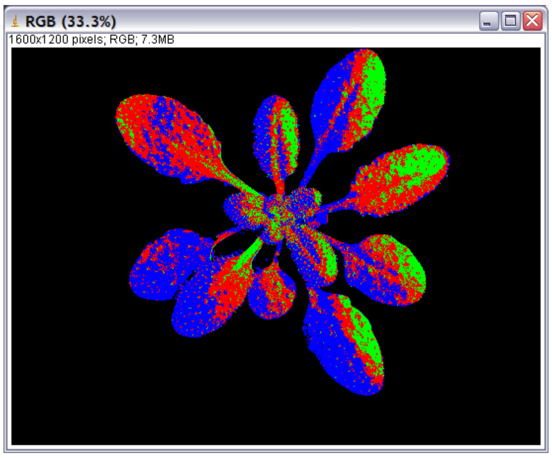


Illustration 135: Image segmented into four classes: 1. background - black, 2. dark areas - blue, 3. brighter areas - read, brightest areas - green.

The k-means clustering works for color images as well. Try it on the image flamingo.png and on one of the images from the plant robot folder. Use a bigger number of classes (about 15) on the later.
The k-means-clustering interprets the intensity values in n-colors as coordinates in an n-dimensional space. Each cluster is represented by its centroid. Centroids are initially randomly set and optimized afterwards. Pixels are grouped by their proximity to a cluster's centroid.

5.4 Watershed segmentation

Open the image *two-circles.tif*. Apply a threshold. Now there are two objects touching each other and we want them to be separated. This can be done using a binary watershed. Apply it from *Process>Binary>Watershed*.

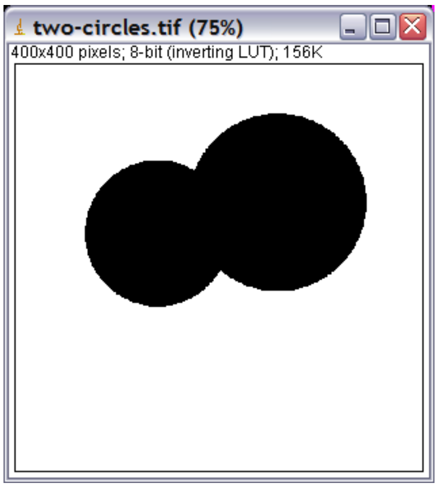


Illustration 136: The thresholded image contains two objects touching each other.

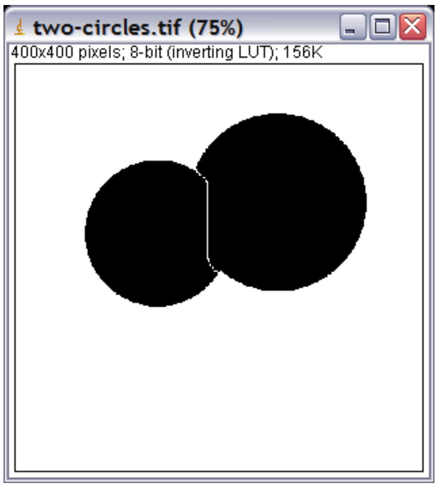


Illustration 137: The binary watershed algorithm separated the two objects.

Starting from the binary image a distance map is computed. This gives a grayscale image where the intensities represent the distance from the border for each object. You can create the distance map from the original image with *Process>Binary>Distance Map*. Imagine the intensity values as height above the ground. This way the image becomes a landscape with mountains in places of high intensities and valleys in places of low intensities. Apply *Analyze>Surface Plot* or use *Interactive 3D surface plot* plugin (*Plugins>3d*) the to see the image this way. The watershed slowly fills the image with water starting from the level 0 and going up one level in each step. In the beginning some separated basins will be created. Whenever the rising water unites two basins a dam is build and remembered.

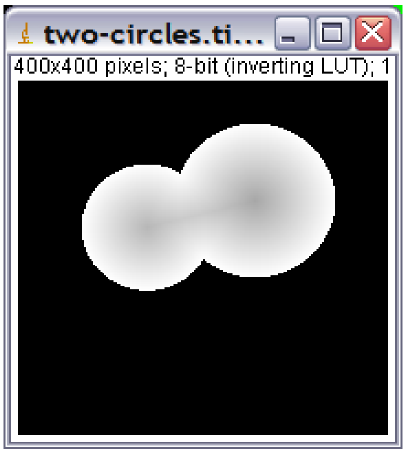


Illustration 138: The distance map, pixels within the object are darker, the further away from the border they are.

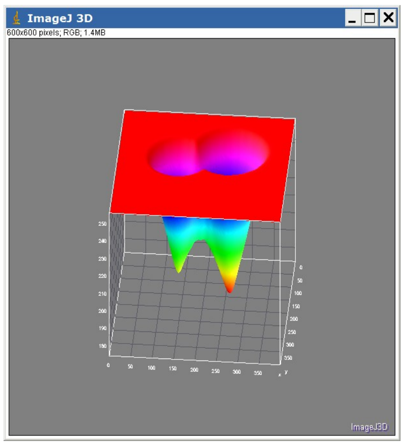


Illustration 139: The distance map as a surface plot.

The watershed can be used with a grayscale image directly. To try this we need to install the watershed plugin from http://bigwww.epfl.ch/sage/soft/watershed/.
Download the zip, copy it into the plugins folder and unzip it. Restart ImageJ. Now let’s try to segment the image compartments-1.jpg with the help of the watershed. Open the image and start the plugin from *Plugins>Watershed>Watershed*. Use a high value for the smoothing. Press the Smooth button. Invert the result image so that the basins are dark, using Edit>Invert or shift+I on the image. Select 8-connected. This means every pixel has 8 neighbors, 2 above, 2 below and the 4 corners. Run the watershed.

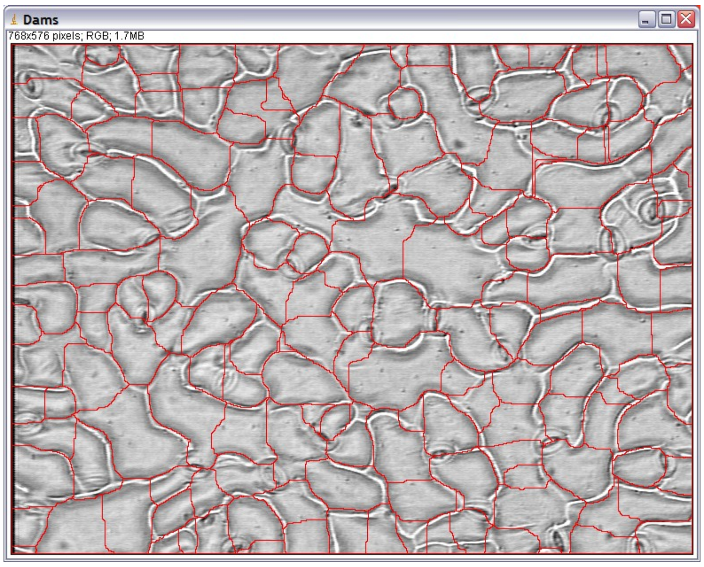


Illustration 140: The red lines represent the dams found by the watershed. The image is slightly oversegmented.

Can you get a better result by applying an edge enhancing filter? Try Process>Find Edges or shift+f on the image before using the watershed.

6. Image types and formats

6.1 Spatial resolution

The optical resolution is defined as the shortest distance between two points on a specimen that can still be distinguished by the observer or camera system. For a microscope the resolution depends on the wavelength and the numerical aperture of the objective. The resolution is the wavelength divided by 2 times the numerical aperture. Smaller wavelength and higher numerical aperture yield higher resolution.
The pixel size in an image taken with a microscope and a ccd camera is the product of the size of a single ccd element multiplied by the binning used and divided by the overall magnification factor.
The more pixels an image has, the higher can be the resolution, though the number of pixels alone doesn't tell anything about the actual resolution. The more pixels an image has, the larger will be the file size in the same time. By how much the file size grows with the number of pixels, depends on the bit-depth of the image. Eight-bit images store intensity values between 0 and 255. Each pixel is stored in 8 bits or one byte. If the image has 400x400 pixels the size will be 400*400 = 160000 bytes or about 156kb (1kb = 1024b). If we double the size of the image to 800x800 pixels the file size will be four times as large, i.e. 625kb. If we take four times the pixels in x and y direction the file size will be 16 times as large.

6.2 Image types

In ImageJ there are the following basic image types available:

- 8 bit grayscale images (values from 0 to 255)
- 16 bit grayscale images (values from 0 to 65535)
- 32 bit floating point number images (uses signed floating point numbers)
- 8-bit color (values from 0 to 255), an 8 bit image with a lookup table
- RGB Color (3 channels of values from 0 to 255, bit depth 24)

What happens if we convert an 8-bit grayscale image into a 16-bit grayscale image? Try it! Open the image cells2.tif and look at the histogram (press h on the image). Then convert it to 16bit using Image>Type>16 bit. Display the histogram again. As you see the intensity values have not been changed.

6.2.1 Conversion from 16 to 8 bit

Open the image PAX_Noc30.tif now. Display the histogram. This is a 16 bit image containing values from 1000 to 35678. What will happen when we convert it to 8bit? Use *Image>Type>8 bit*. The values have been scaled down from the min/max values 1000 and 35678 to 0 and 255.

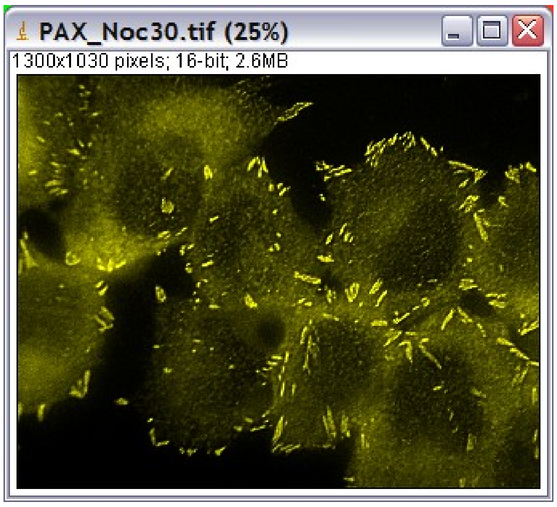
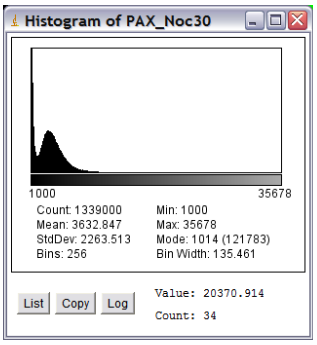


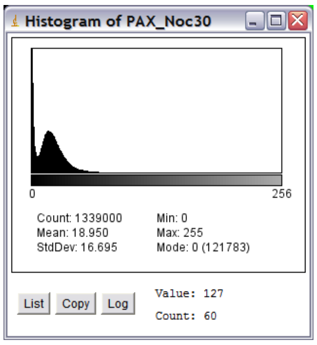
Illustration 141: The 16-bit images contains intensity values from 1000 to 35678.



```
Count: 1339000      Min: 1000
Mean: 3632.847      Max: 35678
StdDev: 2263.513    Mode: 1014 (121783)
Bins: 256           Bin Width: 135.461

Value: 20370.914
Count: 34
```

Illustration 142: The histogram of the 16-bit image.



```
Count: 1339000      Min: 0
Mean: 18.950        Max: 255
StdDev: 16.695      Mode: 0 (121783)

Value: 127
Count: 60
```

Illustration 143: The histogram after conversion to 8-bit.

In *Edit>Options>Conversions* you can turn the scaling off. However then all values above 255 will simply be truncated when converting to 8 bit. The result image will be all white, since the smallest value in the original image is 1000.
Compare the histograms of the results of

- Converting the 16 bit image to 8 bit
- Use *Process>Enhance Contrast>Normalize* on the 16 bit image before converting to 8 bit.
- Convert the 16 bit image to 8 bit and do the normalization

6.2.2 RGB Images

Open an image from the plant robot folder. For color images there is a profile tool as well. Draw a line across the plant and the earth and call *Plugins>Color Functions>RGB Profiler*.

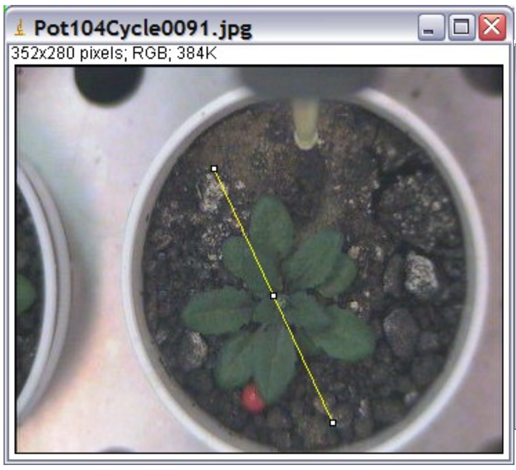


Illustration 144: An RGB-color image.

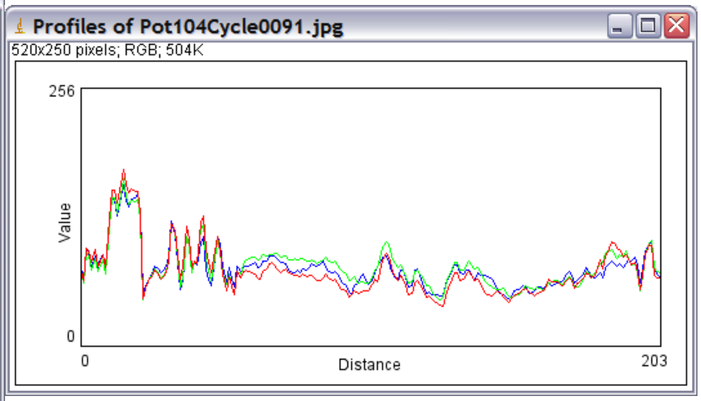


Illustration 145: The color profile along the line.

An RGB image can be split into its three components with *Image>Color>RGB Split* and two or three 8-bit grayscale images can be combined into an RGB color image.
Besides this, an RGB image can be converted into a RGB stack or into a HSB stack using the type menu. HSB stands for hue, saturation and brightness.

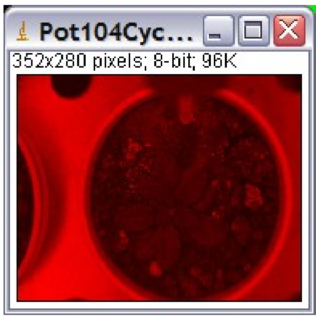


Illustration 146: Red component of the image.

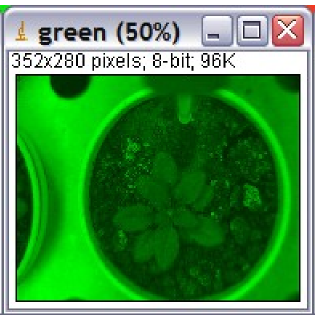


Illustration 147: Green component of the image.

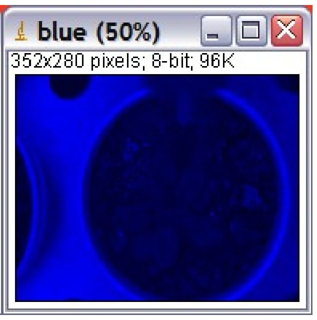


Illustration 148: Blue component of the image.

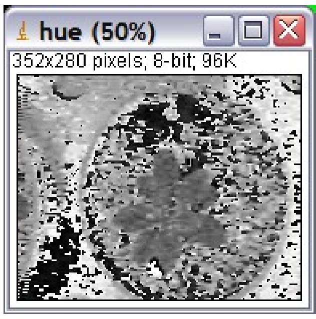


Illustration 149: Hue component of the image.

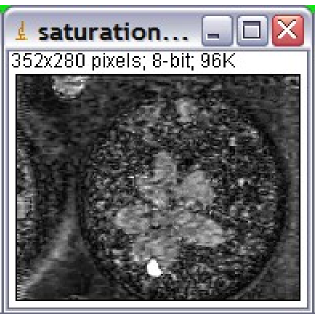


Illustration 150: Saturation component of the image.

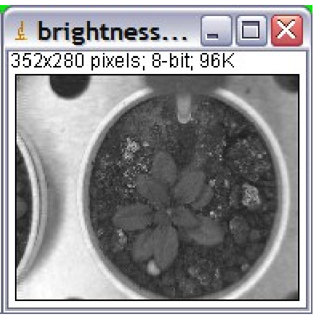


Illustration 151: Brightness component of the image.

An RGB image can be converted into an indexed color image. You can choose the number of colors that will be used and a lookup table will be created accordingly.
Create two images of same size and draw a white rectangle on a dark ground into the two images in a way that the two rectangles are overlapping when the images are combined. Apply rgb merge and put the images into the red and into the green channels. What will be the color of the overlapping region?

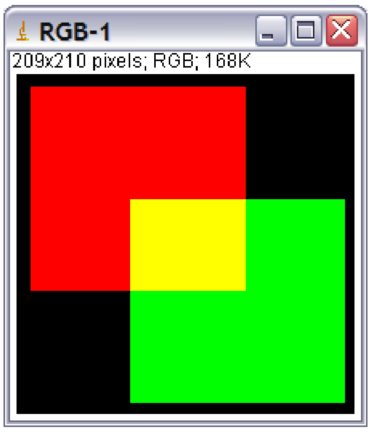


Illustration 152: Green and red mix to yellow in an additive color model.

6.2.3 File Formats

Open an image and save it as jpg, using *File>SaveAs>Jpeg*. Open the jpg file, and calculate the difference between the jpg version and the original image. What has happened? If you look at the file sizes you'll see that the jpg version is smaller than the tiff version. Jpg is a compressed format. There are lossless and not lossless compressions. Jpg uses the first scheme. Some information is sacrificed in order to achieve a smaller image in which the difference is not too much visible for human beings. This is similar to mp3 audio compression.

7. Image Analysis

For our purpose we can define the term image analysis in the following way:

- *Image analysis is the extraction of meaningful information from digital images by means of digital image processing techniques.*

7.1 Counting objects

7.1.1 Counting objects manually

Open the image *nuclei.tif*. How many nuclei are there in the image? The cell counter plugin, written by Kurt De Vos helps to count objects manually. Open the cell counter from *Plugins>Paricle Analysis>Cell Counter*. You first have to press the initialize button to activate the cell counter on the current image. You can count different classes of objects in the same image. Select a counter type and click on a nucleus in the image. The nucleus will be marked with a number corresponding to the selected class and the count of the class will be incremented by one. If you select delete mode a click deletes the nearest marker of the selected class and decrements the counter of that class.
Once you marked all objects you can use results to get a table with the count for the different classes. You can save the markers independent from the image and load them later on again, either on the same, or on another image. Export image creates an image where the markers are drawn into the image. You can save this image to show which cells you counted, however in

contrast to saving the markers, you can't continue the counting later on with the exported image.

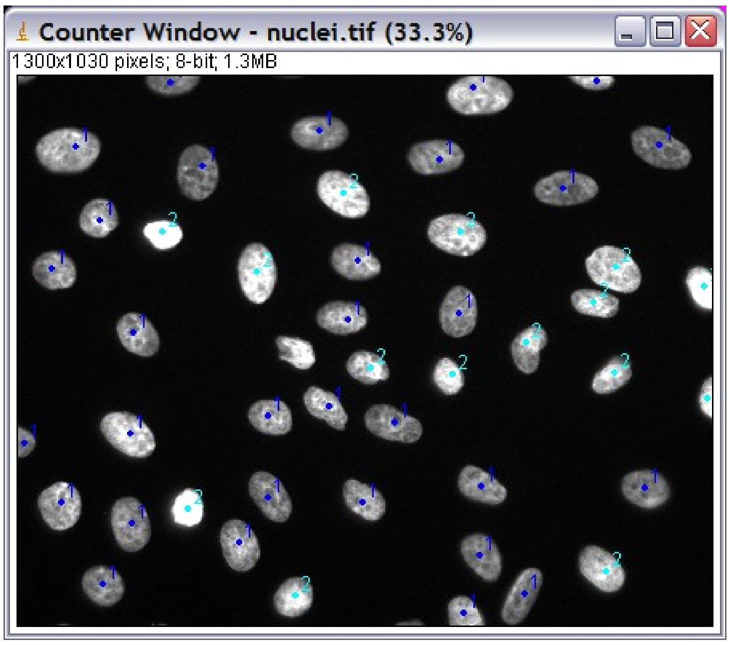
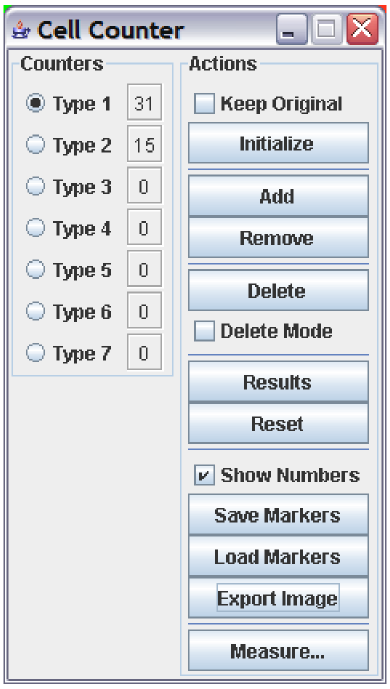

Counter Window - nuclei.tif (33.3%)
1300x1030 pixels; 8-bit; 1.3MB

Illustration 153: Nuclei marked with the cell counter tool.



Counters	
● Type 1	31
○ Type 2	15
○ Type 3	0
○ Type 4	0
○ Type 5	0
○ Type 6	0
○ Type 7	0

☐ Keep Original
☐ Delete Mode
☑ Show Numbers

Illustration 154: The cell counter tool.

7.1.1 Counting and measuring separated objects automatically

Use the threshold adjuster to create a mask from the nuclei image. Then go to Analyze>Set Measurements and make sure that centroids is selected. Open the Particle Analyzer from *Analyze>Analyze Particles* or from the toolbox.

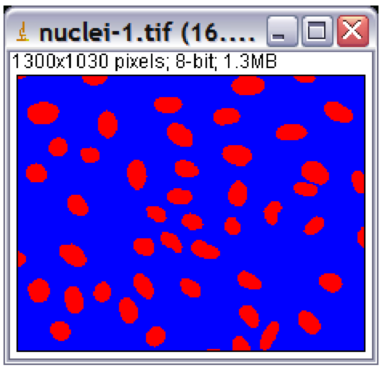
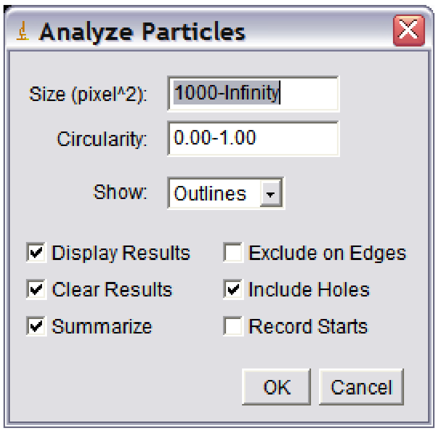


Illustration 155: The mask created with the tresjold adjuster.



Size (pixel^2): 1000-Infinity
Circularity: 0.00-1.00
Show: Outlines

☑ Display Results ☐ Exclude on Edges
☑ Clear Results ☑ Include Holes
☑ Summarize ☐ Record Starts

Illustration 156: The particle analyzer.

You can exclude particles by their size and circularity. Put in a minimum size of 1000. Select *outlines* in the show field and select *Display Result*, *Summarize* and *Include Holes*. As a result you get a results table with the measurements for each particle, a results table with the summary and an image of the numbered outlines of the particles taken into account. You find the total number of objects in the summary results table.

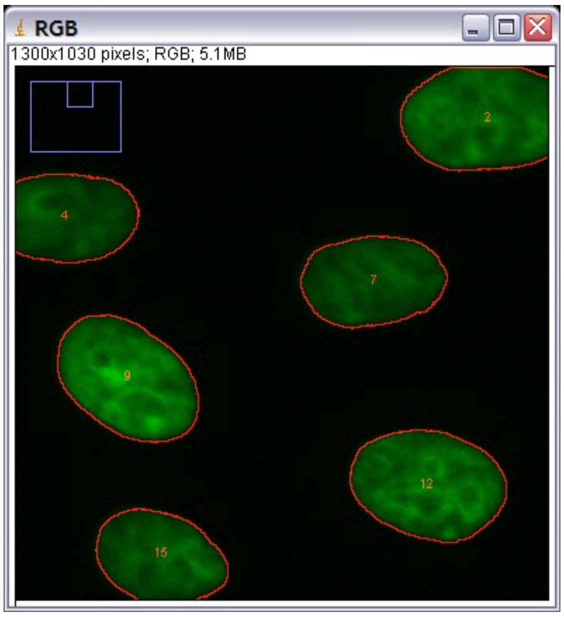
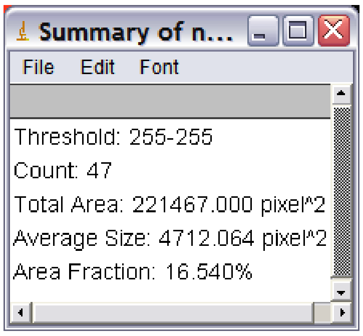


Illustration 157: Detail of the overlay of the original image and the outlines of the particles taken into account by the particle analyzer.



```
Threshold: 255-255
Count: 47
Total Area: 221467.000 pixel^2
Average Size: 4712.064 pixel^2
Area Fraction: 16.540%
```

Illustration 158: The summary of the measurements includes the total count.

As an alternative to using the outlines to mark which particles have been taken into account, you can use the *Analyze>Label* command to stick a number to each object. This is only possible if you configured the particle analyzer to measure the centroids of particles.

7.1.2 Counting and measuring objects touching each other automatically

7.1.2.1 Statistical approach

It can be difficult to separate objects touching each other. One approach to count them is not to try to separate them, but to measure the total area and to divide the result by the average area of a particle. Open the image *si3_hoechst_1.tif* from the count cells folder. Convert the image to 8 bit and apply a threshold. To estimate the average size of a particle use the wand tool to select singular particles and measure their area. You can use *Analyze>distribution* to get the average of a column of the results table. Then use the particle analyzer to measure the total area of all particles. In case you don't need to exclude small particles that don't represent cells, you can use *Edit>Selection>Create Selection* and measure the selected area.

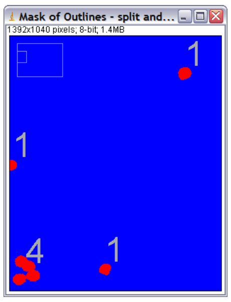

Mask of Outlines - split and...
1392x1040 pixels; 8-bit; 1.4MB
1
1
4
1

Illustration 159: Particles bigger then the average pariticle size are counted multiple times.

7.1.2.2 Local Threshold

Sometimes unsharp masking can help to separate particles. Try the local threshold plugin from *Plugins>Segmentation>Local Threshold* on the image *hst-0.tif* in the cells in structures folder. The plugin applies a median filter and subtracts the result from the original image. The result image is a 32 bit image. It then lets the user choose a threshold value.

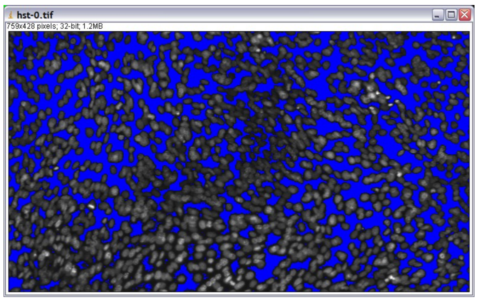


Illustration 160: The original image with a threshold.

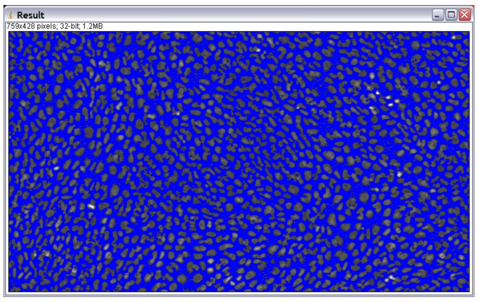


Illustration 161: The image after application of the local threshold plugin.

7.1.2.3 Watershed

We can use the greyscale watershed plugin to separate the cells in the *hst-0.tif* image. Invert the image, so that the nuclei are dark and the space around them is bright. Use a small radius for the smoothing filter. Select the show the watershed dams only option. Run the watershed. Invert the watershed image and subtract it from the original *hst-0.tif* image. The borders between the cells are now set to zero and you can use the threshold adjuster to get a mask for the separated cells.

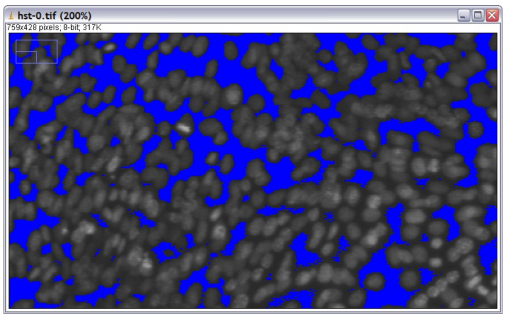


Illustration 162: The original image.

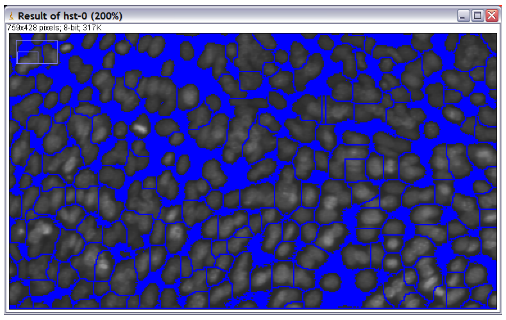


Illustration 163: Cells separated by the watershed plugin.

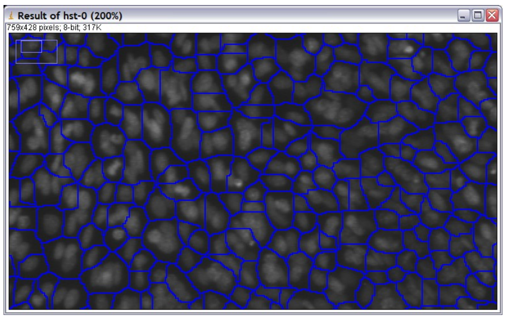


Illustration 164: Image with watershed dams.

7.1.2.4 Other methods

Other methods use for example first or second order gradients of the image or mathematical morphology.

7.2 Comparing intensities

Open the image *A4 Rhod 1.tif*. We want to compare the total intensity in the nuclei with the total intensity in the cytoplasm. The image *A4 dapi 1.tif* helps us to find the nuclei. Apply the *hilo* lut to the rhod image and subtract a baseline value, so that the empty background becomes zero. Threshold the dapi image. If there are holes in the mask of the nuclei use Process>Binary>Fill Holes to close them. Create a selection from the mask and if necessary modify the selection, so that it only contains the nuclei. Transfer the selection from the mask image to the rhod image using *Edit>Selection>restore* or *CTRL+Shift+e*. Configure the measurements to include the total intensity. And measure the intensity in the selection (*ctrl+m*). Inverse the selection (*Edit>Selection>Make Inverse*) and measure again.

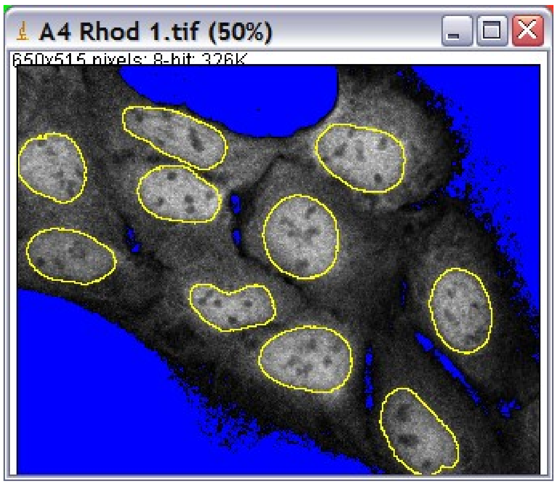


Illustration 165: Measuring the intensity in the nuclei and in the cytoplasm.

7.3 Classifying objects

Sometimes you might not only want to separate objects from the background and from each other but also to distinguish between different kinds of objects in the image. One could for example be interested in distinguishing between normal and apoptotic cells. To do so you need to find a set of features that separates the different classes of objects you are looking for.

Load the image ovals and triangles and try to separate the ovals from the triangles. In this case the two classes can be distinguished by the feature roundness. Use the particle analyzer to separate the two classes of objects.

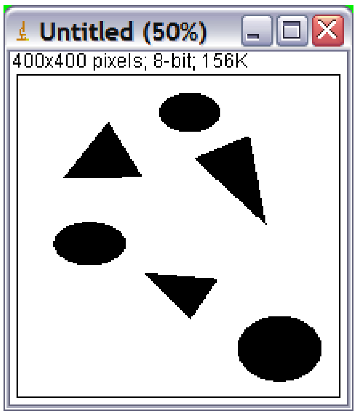


Illustration 166: Different kinds of objects in one image.

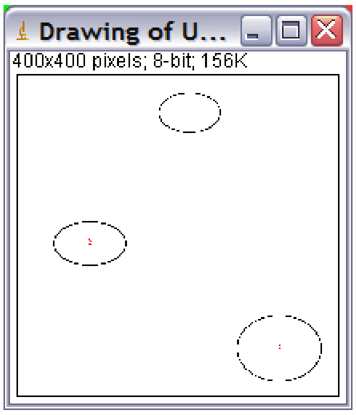


Illustration 167: Ovals have a high roundness.

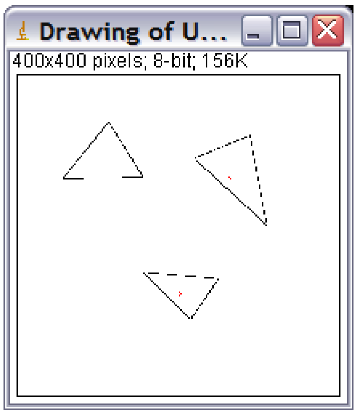
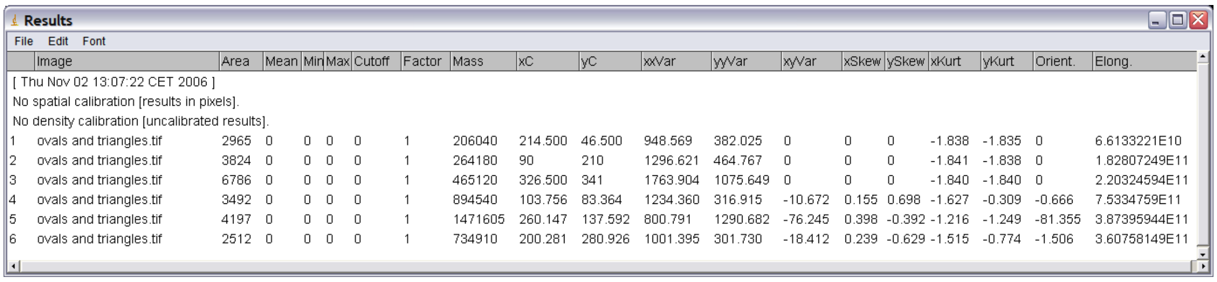


Illustration 168: Triangles have a low roundness.

In general more then one feature might be necessary to classify objects. One useful set of features are the moments of objects and values derived from moments of objects. You can use the Moment Calculator plugin (*Plugins>Particle Analysis>Moment Calculator*) to compute them.



Results

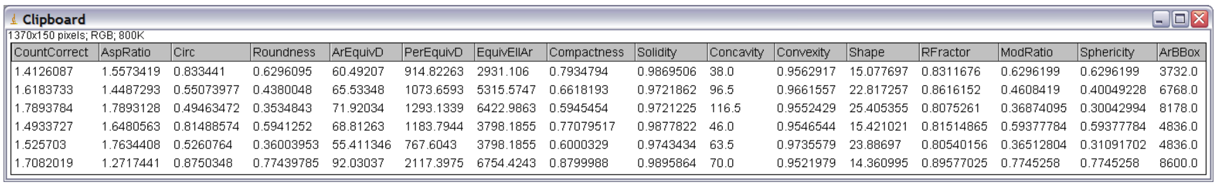
File Edit Font

[Thu Nov 02 13:07:22 CET 2006]
No spatial calibration [results in pixels].
No density calibration [uncalibrated results].

	Image	Area	Mean	Min	Max	Cutoff	Factor	Mass	xC	yC	xxVar	yyVar	xyVar	xSkew	ySkew	xKurt	yKurt	Orient.	Elong.
1	ovals and triangles.tif	2965	0	0	0	0	1	206040	214.500	46.500	948.569	382.025	0	0	0	-1.838	-1.835	0	6.6133221E10
2	ovals and triangles.tif	3824	0	0	0	0	1	264180	90	210	1296.621	464.767	0	0	0	-1.841	-1.838	0	1.82807249E11
3	ovals and triangles.tif	6786	0	0	0	0	1	465120	326.500	341	1763.904	1075.649	0	0	0	-1.840	-1.840	0	2.20324594E11
4	ovals and triangles.tif	3492	0	0	0	0	1	894540	103.756	83.364	1234.360	316.915	-10.672	0.155	0.698	-1.627	-0.309	-0.666	7.5334759E11
5	ovals and triangles.tif	4197	0	0	0	0	1	1471605	260.147	137.592	800.791	1290.682	-76.245	0.398	-0.392	-1.216	-1.249	-81.355	3.87395944E11
6	ovals and triangles.tif	2512	0	0	0	0	1	734910	200.281	280.926	1001.395	301.730	-18.412	0.239	-0.629	-1.515	-0.774	-1.506	3.60758149E11

Illustration 169: Features based on moments of the circles (first 3 rows) and triangles (last three rows) .

Further features can for example be calculated with the Particle8 Plus or other morphological plugins from Gabriel Landini (*Plugins>morphology>Particles8 Plus*).



Clipboard

1370x150 pixels; RGB; 800K

CountCorrect	AspRatio	Circ	Roundness	ArEquivD	PerEquivD	EquivEllAr	Compactness	Solidity	Concavity	Convexity	Shape	RFractor	ModRatio	Sphericity	ArBBox
1.4126087	1.5573419	0.833441	0.6296095	60.49207	914.82263	2931.106	0.7934794	0.9869506	38.0	0.9562917	15.077697	0.8311676	0.6296199	0.6296199	3732.0
1.6183733	1.4487293	0.55073977	0.4380048	65.53348	1073.6593	5315.5747	0.6618193	0.9721862	96.5	0.9661557	22.817257	0.8616152	0.4608419	0.40049228	6768.0
1.7893784	1.7893128	0.49463472	0.3534843	71.92034	1293.1339	6422.9863	0.5945454	0.9721225	116.5	0.9552429	25.405355	0.8075261	0.36874095	0.30042994	8178.0
1.4933727	1.6480563	0.81488574	0.5941252	68.81263	1183.7944	3798.1855	0.77079517	0.9877822	46.0	0.9546544	15.421021	0.81514865	0.59377784	0.59377784	4836.0
1.525703	1.7634408	0.5260764	0.36003953	55.411346	767.6043	3798.1855	0.6000329	0.9743434	63.5	0.9735579	23.88697	0.80540156	0.36512804	0.31091702	4836.0
1.7082019	1.2717441	0.8750348	0.77439785	92.03037	2117.3975	6754.4243	0.8799988	0.9895864	70.0	0.9521979	14.360995	0.89577025	0.7745258	0.7745258	8600.0

Illustration 170: Further features of the circles (rows 1, 4, 6) and triangles (rows 2, 3, 5).

8. Annotating images

8.1 Scale bar, time stamper and event stamper

Open the image *Pot.tif*. This opens a stack of image which represents a time series. You can cycle through the images with the slider at the bottom of the window. We want to add three annotations to the images: A scale bar indicating the length of a known distance, the time point from the beginning of the series and a marker for images in which the flower appears. To add a scale bar we first have to calibrate the image. We either need to know the pixel size or the length of an object in the image. Let's say that the diameter of the pot is 6cm. Make a line selection from one side of the pot to the opposite side. Then open *Analyze>Set Scale...*. Enter the distance 6 and the unit cm and press ok. When the scale is set ImageJ will use for all measurements, i.e. results will now no longer be in pixel but in *cm*.

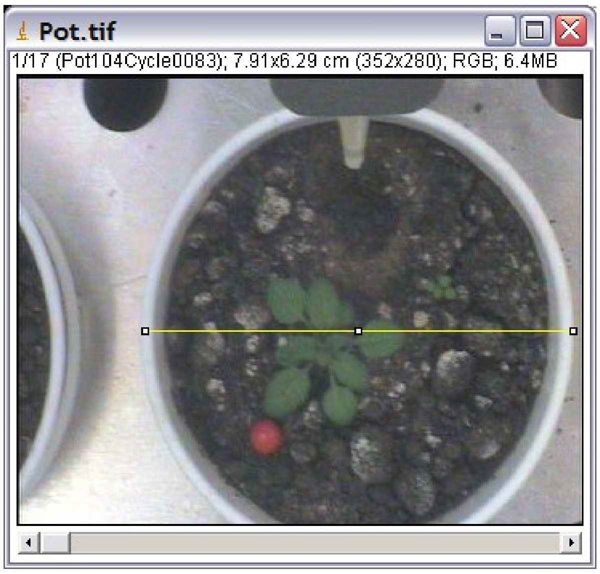
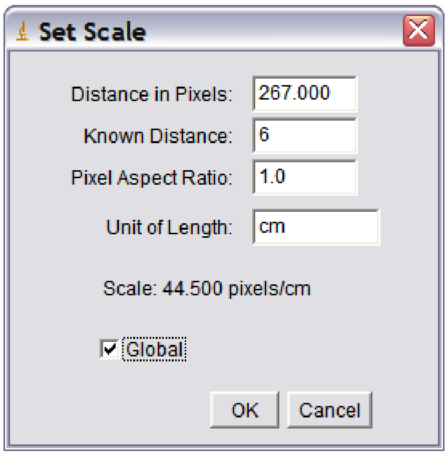


Illustration 171: Selecting a known distance.



Set Scale

Distance in Pixels: 267.000
Known Distance: 6
Pixel Aspect Ratio: 1.0
Unit of Length: cm

Scale: 44.500 pixels/cm

☑ Global

OK Cancel

Illustration 172: Setting the scale.

Go to *Analyze>Tools>Scale Bar* and put a black scale bar of 2cm length into the lower left corner. Select *Label all Slices* to put the scale bar into all images of the series.

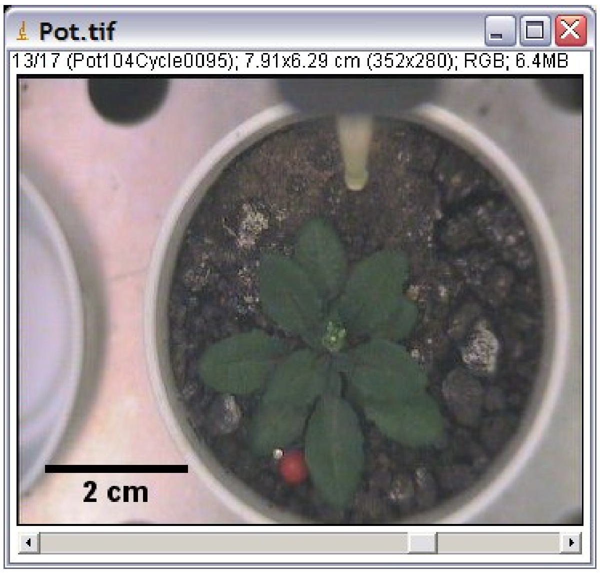
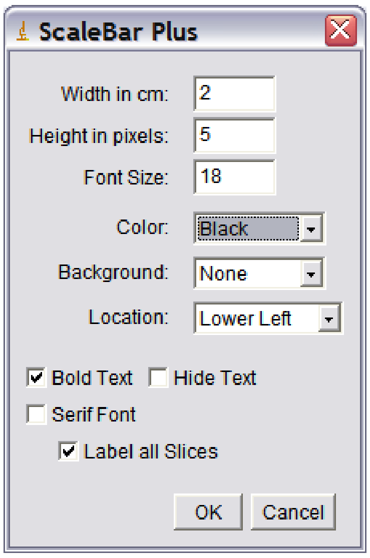


Illustration 173: The black line segment has a length of two centimeter.



ScaleBar Plus

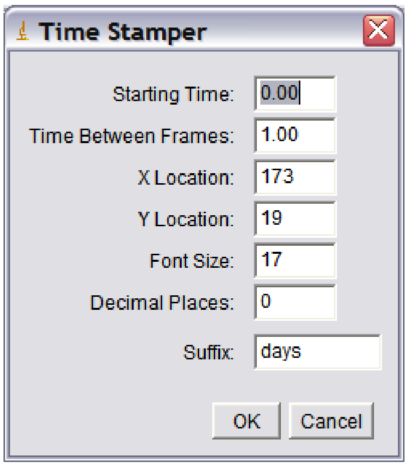
Width in cm: 2
Height in pixels: 5
Font Size: 18
Color: Black
Background: None
Location: Lower Left

☑ Bold Text ☐ Hide Text
☐ Serif Font
☑ Label all Slices

OK Cancel

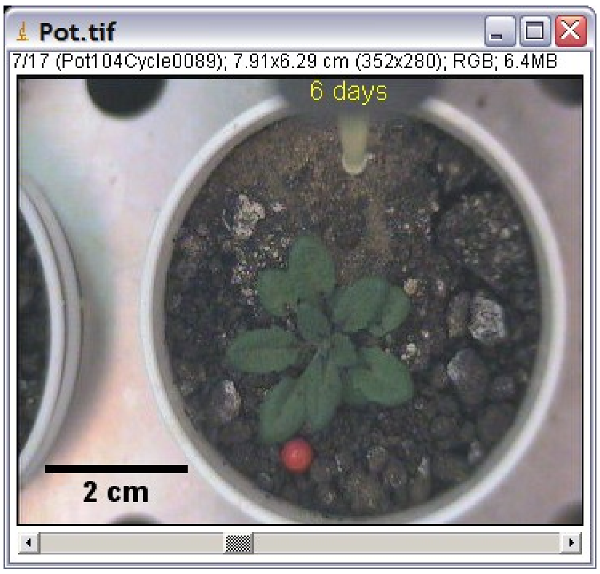
Illustration 174: Adding a scale bar to each image of the series.

We will now add a text indicating the time when the image was made. Let's say the image have been made with an interval of 1 day between to images. Make a rectangular selection in the place where the stamp should appear. Double click the pipette icon and set the foreground color to the color you want the stamp to have. Then open the time stamper from *Plugins>Movie>Time Stamper*.


Time Stamper

Starting Time: 0.00
Time Between Frames: 1.00
X Location: 173
Y Location: 19
Font Size: 17
Decimal Places: 0
Suffix: days

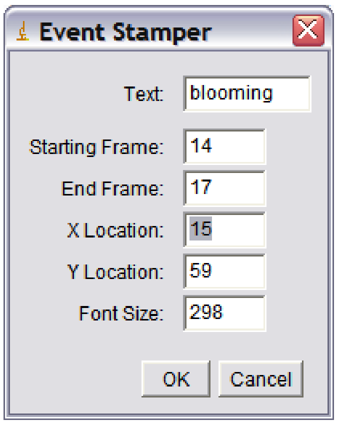
Illustration 175: The time stamper dialog.


Pot.tif

7/17 (Pot104Cycle0089); 7.91x6.29 cm (352x280); RGB; 6.4MB

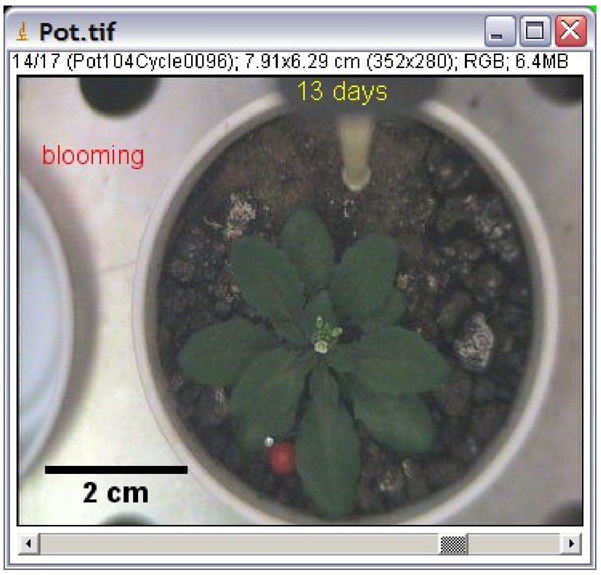
Illustration 176: The time stamp has been added to each image of the series.

Finally add a label *blooming* on the slices 14 to 17. This is done in a similar way as adding the time stamp, only that the command *Plugins>Movie>Event Stamper* is used.


Event Stamper

Text: blooming
Starting Frame: 14
End Frame: 17
X Location: 15
Y Location: 59
Font Size: 298

Illustration 177: The event stamper dialog.


Pot.tif

14/17 (Pot104Cycle0096); 7.91x6.29 cm (352x280); RGB; 6.4MB

Illustration 178: A text has been added to some images of the series.

8.2 Calibration bar

Using a lookup table on a greyscale image, we want to provide a bar that shows which colors represent which intensities, normalized to a range between 0 and 1, in the image.
Open the image *A4 Rhod 1.tif* and apply the lookup table *Rainbow RGB*. There are two ways to achieve our goal. One way is to make a selection in the image and run *Plugin>LUT>Add*

Ramp. This will add a ramp with the colors of the lut from the bottom with the value 0 to the top with the value 255. It only works for 8 bit images.

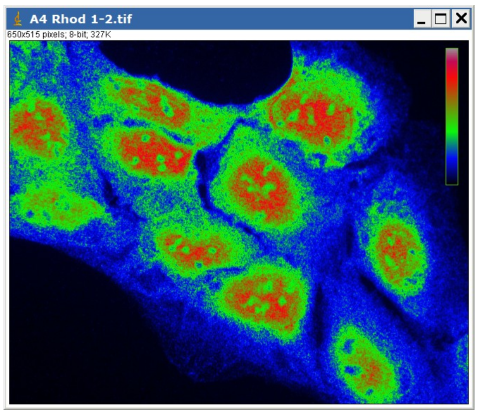
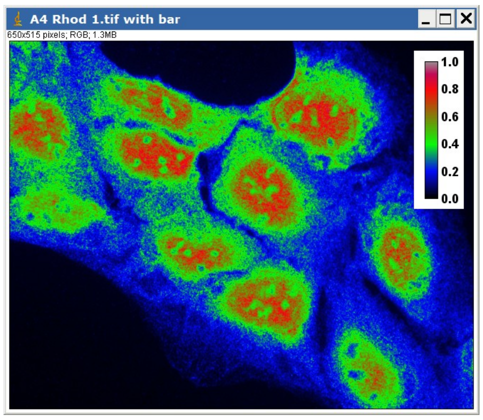


Illustration 179: A ramp shows which color represents which intensity.

The other way is to first calibrate the image intensities using *Analyze>Calibrate...* A calibration bar can then be stamped into the image using *Analyze>Tools>Calibration Bar...* To calibrate the image you have to enter a number of pairs of uncalibrated and calibrated values. You can select the form of a curve that will be fitted to the entered data. For a calibrated image, intensity values will be displayed in the calibrated form, followed by the greylevel value in brackets. Measurements will be in calibrated form.
Call *Analyze>Calibrate...* and enter the points 0, 0 and 255, 1. Select the function straight line. Then use *Analyze>Tools>Calibration Bar...* stamp the calibration bar into the image.



9. Working with multi-dimensional data

In general image data can have multiple dimensions. 3 spacial dimensions, the time dimension and many channels (for example different wavelengths).

9.1 Time series

Open the image *pot.tif*. You can play the image sequence using *Image>Stacks>Start Animation*. You can use the last button in the ImageJ launcher window to switch the current tool set to the stack tools. This will display buttons to:

- animate the stack (click again ti stop),
- go to first and last slice,
- go one slice forwards / backwards
- add a slice after the current slice / delete current slice

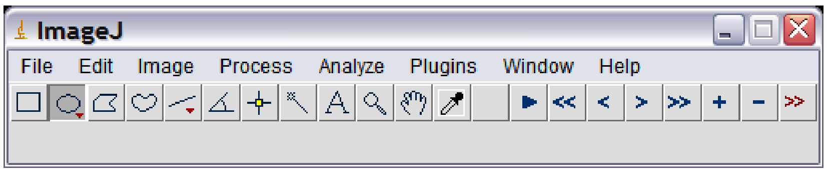


Illustration 180: ImageJ with the stack tool set selected.

As you see the camera position is not exactly the same for each image which makes the plant somewhat jump around. Use *Plugins>Stack Shuffling>Align* slices for an automatic stack registration. Choose either rigid body or translation as transformation.
Make a rectangular selection, so that the black borders that have been created by the stack registration are outside of the selection and duplicate the stack (*ctrl+shift+d*).
You can export the series as a movie in one of the following formats:

- animated gif
- avi movie
- quicktime movie

Use *File>Save As* and one of these formats to export the series as a movie. If your image sequence is to big to fit into memory at once, you can still create a movie from it by using the virtual stack opener (*File>Import>Virtual Stack*).

9.1.1 Measuring velocities

Open the image moving-particles. You see two circles that move from the lower left to the upper left side of the image. We want to measure the lengths of the paths they traveled.

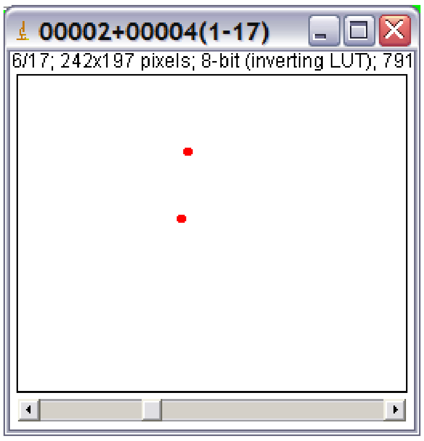


Illustration 181: Two moving particles.

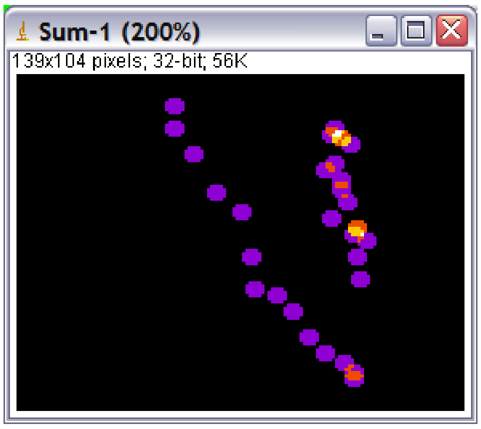
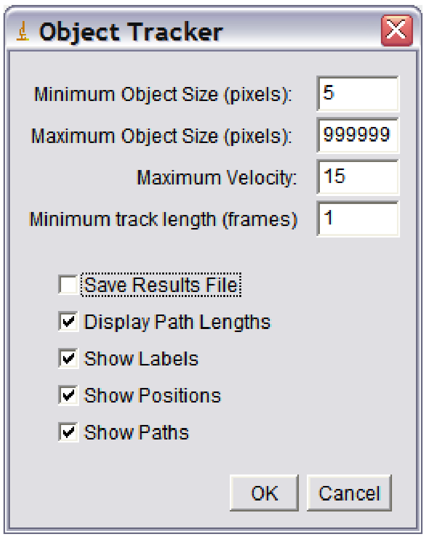


Illustration 182: A projection of the time series.

In principal this could be done manually using for example the polygon selection tool. Go to the first slice set the first polygon point to the into the first particle, then go one slice ahead using the '>' key and set the next point and so on. When the path is finished press *ctrl+m* to measure its length.
Another possibility is to create a projection of the stack (*Image>Stack>Z Project...*) and use the polygon selection tool on it.
The *Mtrack2* plugin measures the track length automatically. Open it from *Plugins>Particle Analysis>Mtrack2*.



Minimum Object Size (pixels): 5
Maximum Object Size (pixels): 999999
Maximum Velocity: 15
Minimum track length (frames) 1

- [ ] Save Results File
- [x] Display Path Lengths
- [x] Show Labels
- [x] Show Positions
- [x] Show Paths

Illustration 183: The MTrack2 plugin.

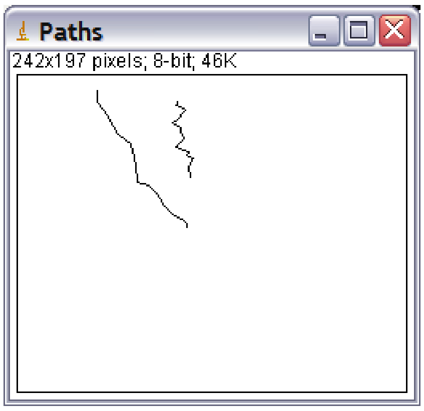
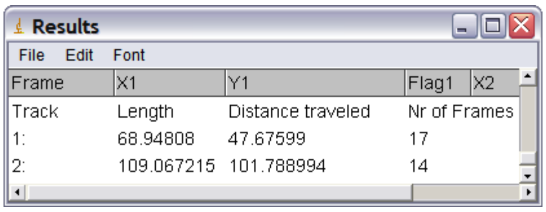


Illustration 184: The paths found by the MTrack2 plugin.



Frame	X1	Y1	Flag1	X2
Track	Length	Distance traveled	Nr of Frames	
1:	68.94808	47.67599	17	
2:	109.067215	101.788994	14	

Illustration 185: The plugin answers the distance travelled by the particles (length) and the linear distance from the start point to the end point of the track.

10. Biological applications

10.1 Colocalization analysis

10.1.1 Visualizing colocalization using overlays

In principle an overlay of the two channels, using two appropriate lookup-tables, like for example red and green or cyan and magenta, should give as a hint if two fluochromes are in the same place. In the red/green case places in which the red and green intensities are close to each other will appear yellow in the image. Another possibility is to use magenta, green and blue.

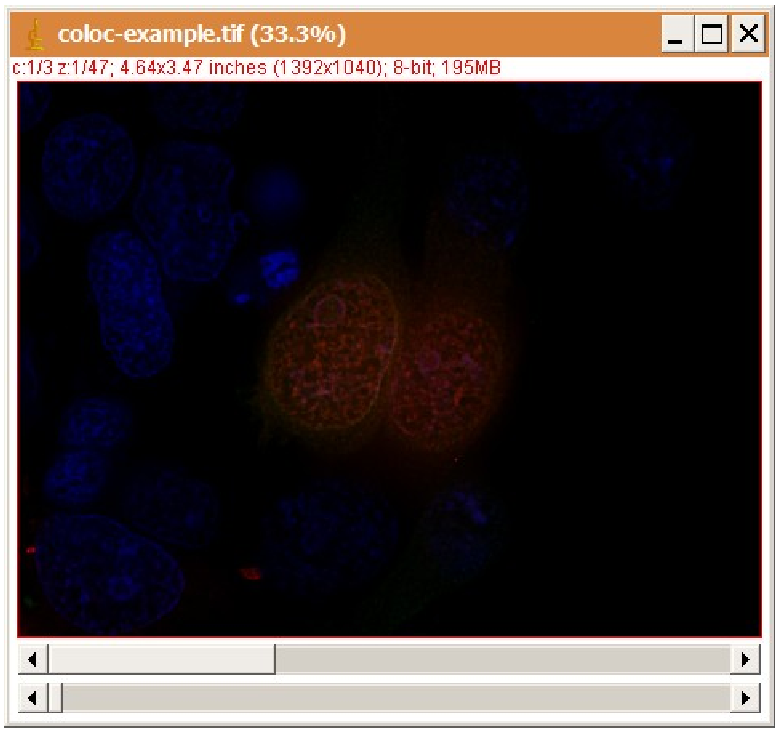
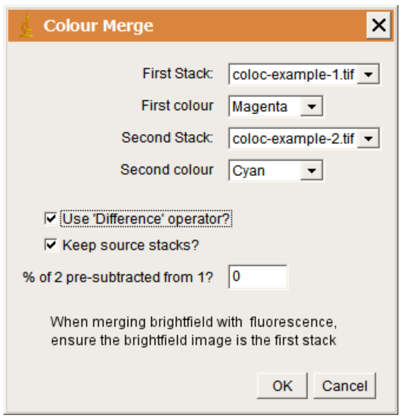

coloc-example.tif (33.3%)
c:1/3 z:1/47; 4.64x3.47 inches (1392x1040); 8-bit; 195MB

Illustration 186: The overlay of the channels. The green channel has lower intensities and is invisible.

But there are some problems with this approach. Yellow spots will only become visible if the intensities in both channels are similar, both channels must have similar histograms. Histogram normalization can be used but may suggest false results since intensities don't represent the real quantities of molecules anymore.
Open the images *coloc-Rhod*, *coloc-GFP* and *coloc-Hoechst* from the coloc folder. Create an overlay of the channels that shows the colocalization between gfp and rhodamine.
Instead of an additive overlay of channels the diference can be used. Apply magenta and cyan lookup tables, convert the images to RGB and calculate the difference using the image calculator. The same result can be achieved using the Colour merge command from the menu *Plugins>Colour functions*. Check *Use difference operator* in the dialog.

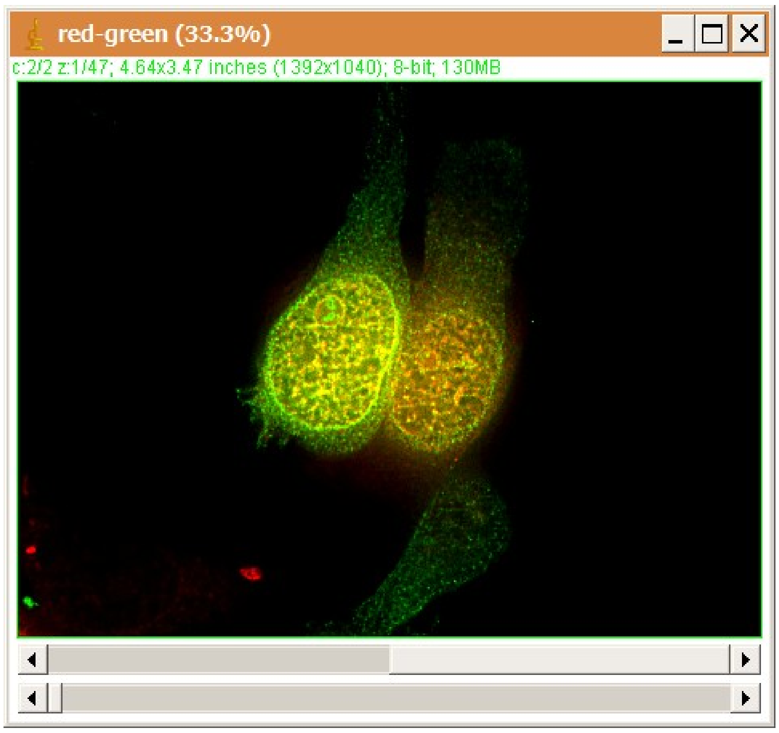

Colour Merge

First Stack: coloc-example-1.tif
First colour: Magenta
Second Stack: coloc-example-2.tif
Second colour: Cyan

☑ Use 'Difference' operator?
☑ Keep source stacks?
% of 2 pre-subtracted from 1? 0

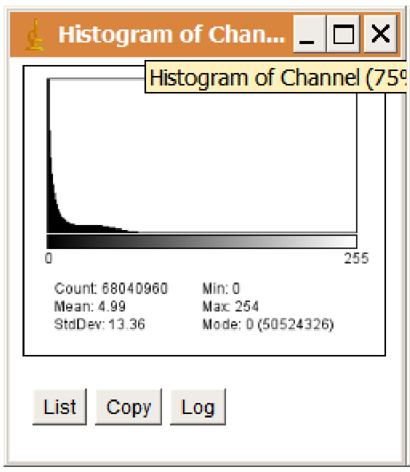
When merging brightfield with fluorescence,
ensure the brightfield image is the first stack

Illustration 187: The color merge dialog.


red-green (33.3%)

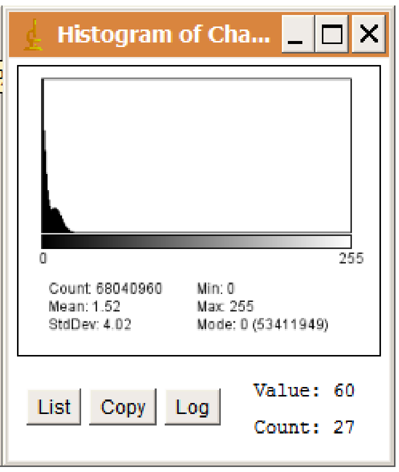
c:2/2 z:1/47; 4.64x3.47 inches (1392x1040); 8-bit; 130MB

Illustration 188: Overlay of the red and green channel after histogram normalization.



```
Count: 68040960     Min: 0
Mean: 4.99          Max: 254
StdDev: 13.36       Mode: 0 (50524326)
```

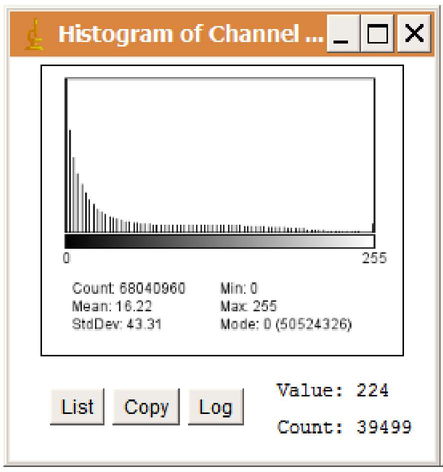
Illustration 189: The histogram of the red channel.



```
Count: 68040960     Min: 0
Mean: 1.52          Max: 255
StdDev: 4.02        Mode: 0 (53411949)
```

```
Value: 60
Count: 27
```

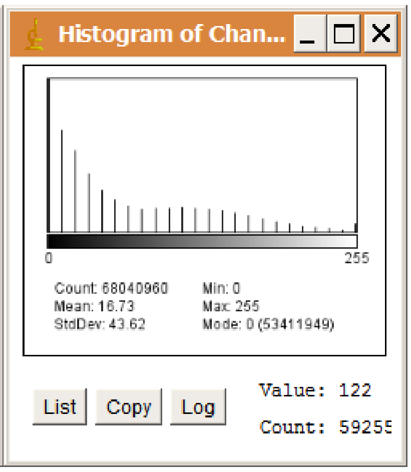
Illustration 190: The histogram of the green channel.



```
Count: 68040960     Min: 0
Mean: 16.22         Max: 255
StdDev: 43.31       Mode: 0 (50524326)
```

```
Value: 224
Count: 39499
```

Illustration 191: The histogram of the red channel after normalization.



```
Count: 68040960     Min: 0
Mean: 16.73         Max: 255
StdDev: 43.62       Mode: 0 (53411949)
```

```
Value: 122
Count: 5925
```

Illustration 192: The histogram of the green channel after normalization.

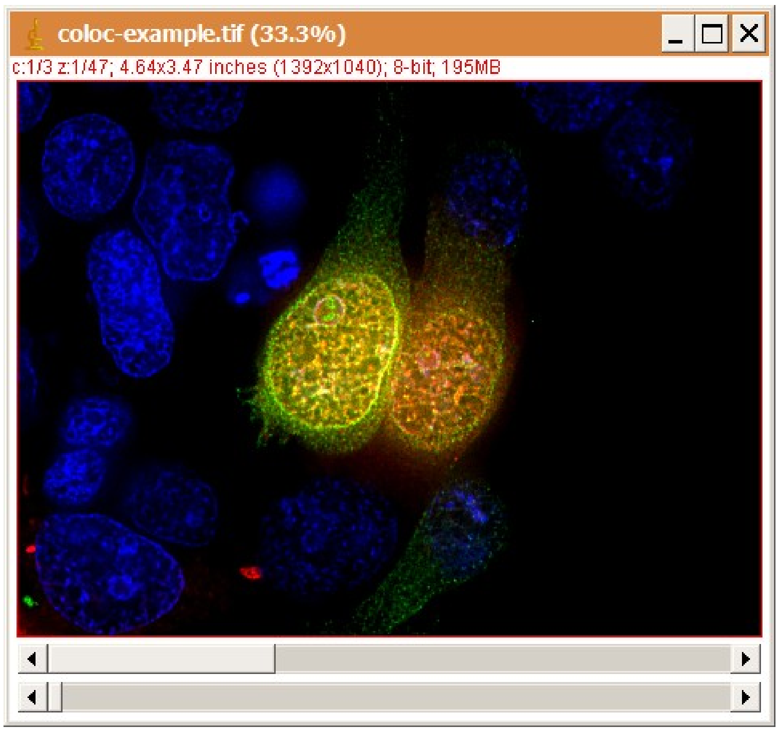

coloc-example.tif (33.3%)
c:1/3 z:1/47; 4.64x3.47 inches (1392x1040); 8-bit; 195MB

Illustration 193: Original overlay with auto-display-adjustment.

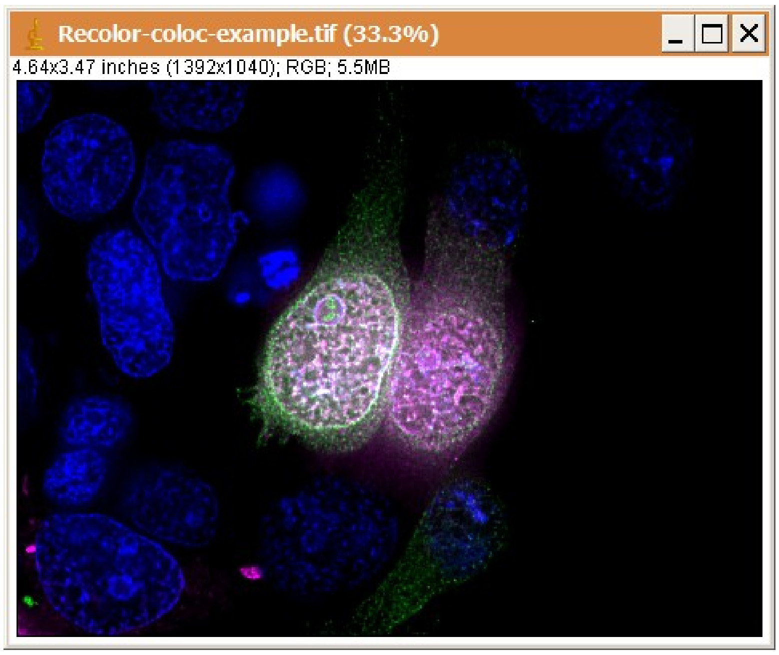

Recolor-coloc-example.tif (33.3%)
4.64x3.47 inches (1392x1040); RGB; 5.5MB

Illustration 194: Original overlay with auto-display adjustment in Magenta/Green/Blue.

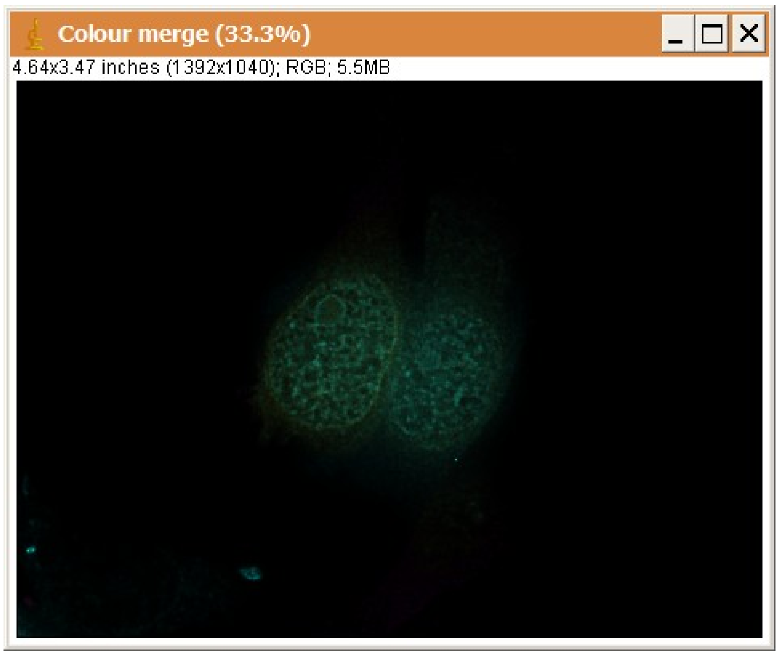


Illustration 195: Color merge with difference of the original channels.

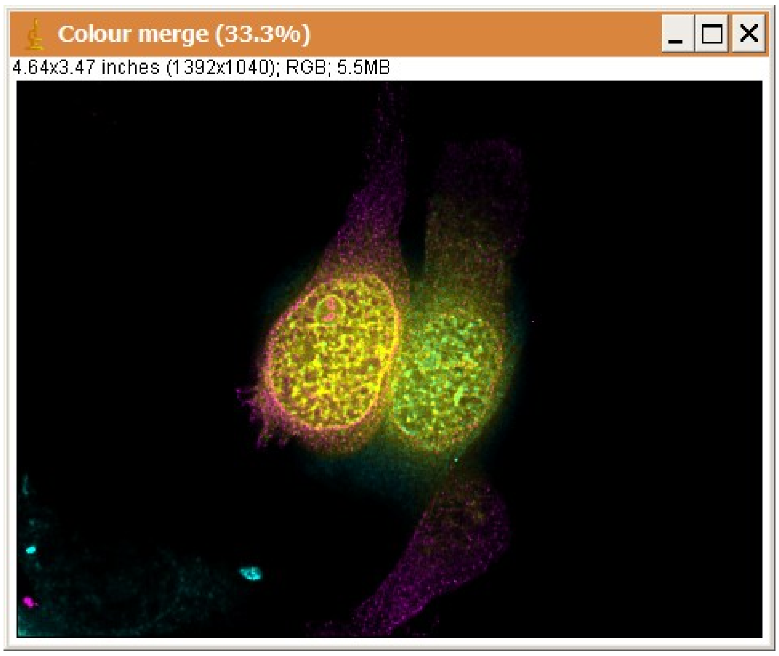


Illustration 196: Color merge with difference after histogram normalization.

10.1.2 Quantification of colocalization

10.1.2.1 Intensity correlation coefficient based methods

In this approach a statistical analysis of the correlation of the intensity values in the two channels is performed. This is mostly done by using correlation coefficients that measuring the strength of the linear relationship between two variables, the intensities in the two channels.

10.1.2.1.1 Pearson's coefficient and scatter plot

A fluorogram or scatter plot can help to judge if colocalization is present or not. In a scatter plot the x-axis represents the intensity values of the pixels in one channel and the y-axis the intensities in a second channel. The pixels of the two images will be compared one by one. If the intensity of pixel (0,0) is 10 in the first image and 15 in the second image, the point 10,15 will be marked in the scatter plot. Sometimes the frequency of the intensity pair is coded by the color or brightness in the scatter plot. A line will be fitted through the points in the scatter plot. A positive slope indicates a correlation, a zero slope indicates that the images are uncorrelated and a negative slope indicates a negative correlation (a mutual exclusion).
The Pearson's Coefficient is a measure of the quality of the linear fit. It is obtained by dividing the covariance of the two variables by the product of their standard deviations. Mark that the thickness of the point cloud and the direction of the line have an influence on the PC, but not the slope of the line. The PC will be near 1 for a positive correlation, near zero in case of no correclation and near -1 for a negative correlation.

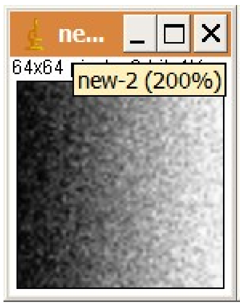


Illustration 197: A noisy image.

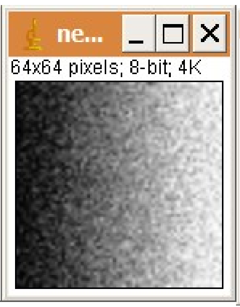


Illustration 198: Another similar noisy image.

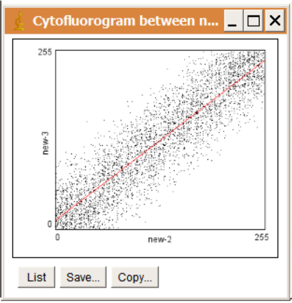


Illustration 199: A scatter plot of correlated images

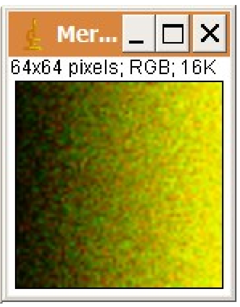


Illustration 200: The color overlay of the input images.

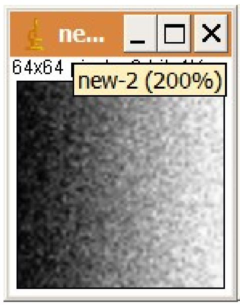


Illustration 201: A noisy image.

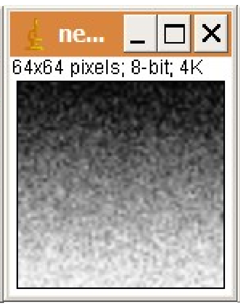


Illustration 202: Another similar noisy image, rotated by 90°.

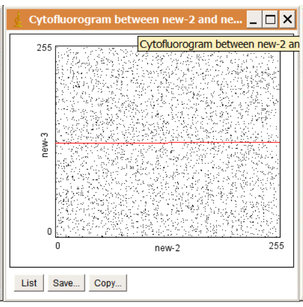


Illustration 203: Scatter plot of uncorrelated images

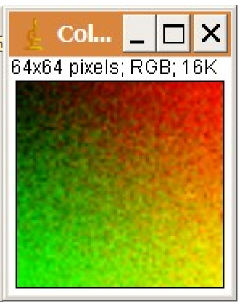


Illustration 204: The color overlay of the input images.

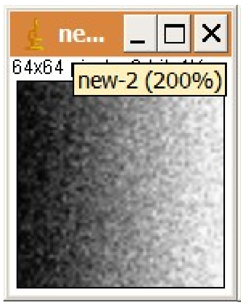


Illustration 205: A noisy image.

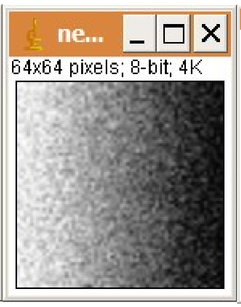


Illustration 206: Another similar noisy image, rotated by 90°.

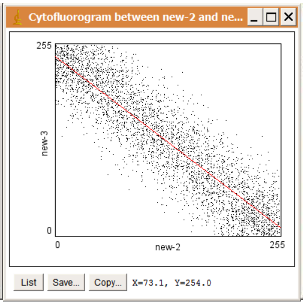


Illustration 207: Scatter plot of anti-correlated images

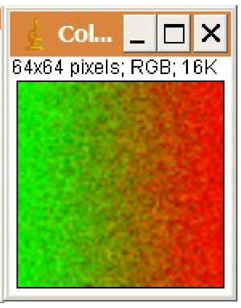


Illustration 208: The color overlay of the input images.

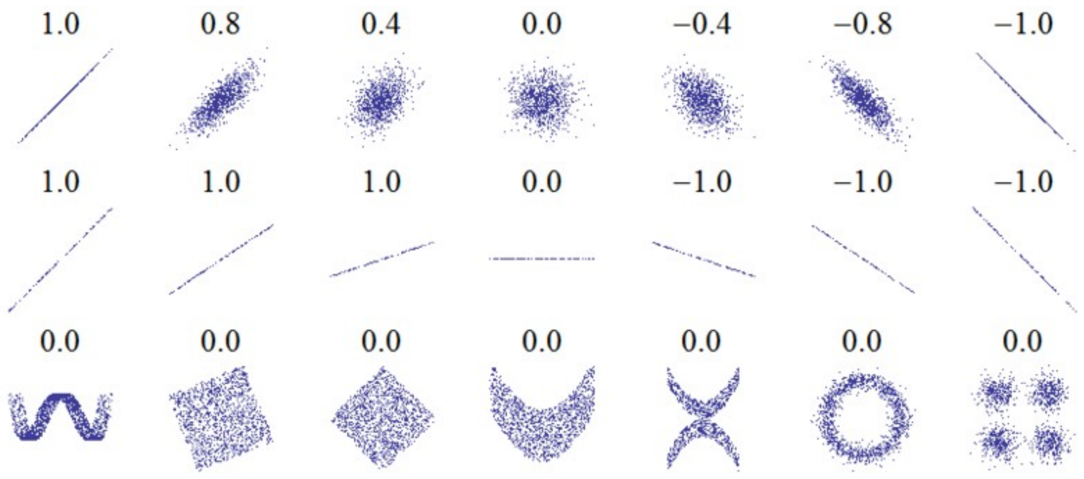


Illustration 209: Pearson's coefficients for different scatter plots.

Open the images *coloc-GFP* and *coloc-Hoechst* from the folder *12 - coloc*. Use the *JACoP* plugin to calculate the scatter plot. Run the plugin from the Plugins menu, select the images in the dialog, select cytofluogram and *Add the zero line*. The scatter plot will be created and the fitted line will be shown. The Pearson's Coefficient will be written to the log window as

Correlation Coefficient. If you want to calculate the Pearson's Coefficient without the scatter plot, check *Pearson's coefficient* instead of *cytofluogram* in the dialog.

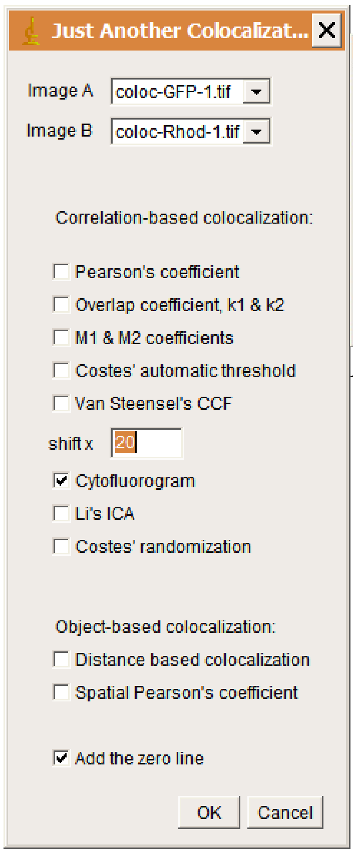

Just Another Colocalizat...

Image A: coloc-GFP-1.tif
Image B: coloc-Rhod-1.tif

Correlation-based colocalization:

☐ Pearson's coefficient
☐ Overlap coefficient, k1 & k2
☐ M1 & M2 coefficients
☐ Costes' automatic threshold
☐ Van Steensel's CCF
shift x: 20
☑ Cytofluorogram
☐ Li's ICA
☐ Costes' randomization

Object-based colocalization:

☐ Distance based colocalization
☐ Spatial Pearson's coefficient

☑ Add the zero line

Illustration 210: The dialog of the JACoP colocalization plugin.

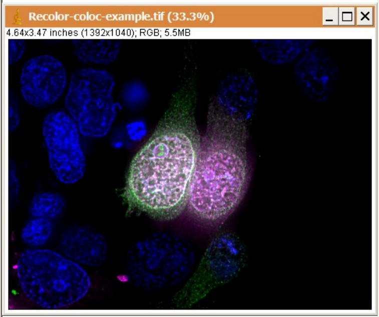

Recolor-coloc-example.tif (33.3%)
4.64x3.47 inches (1392x1040); RGB; 5.5MB

Illustration 211: The overlay of the two input images.

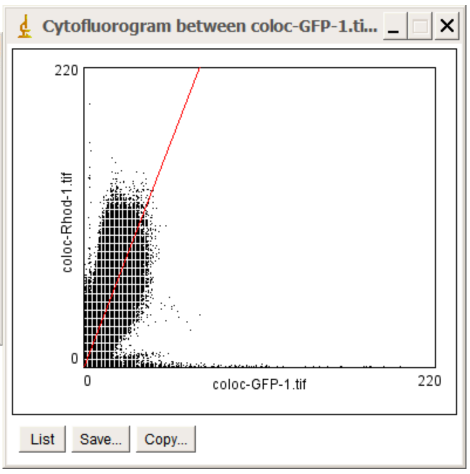


Illustration 212: The scatter plot and the fitted line calculated by the JACoP plugin.

Scatter plot and Pearson's coefficient indicate colocalization, especially when it is complete. However they rarely discriminate differences between partial colocalization. Mid-range coefficients between -0.5 and 0.5 do not allow conclusions to be drawn. The result is dependent on noise and background or the threshold used to supress the background. Bleadthrough of the fluorescents can lead to false results as well. Finally there could be a non linear relationship between the two signals, that could not be detected using this method.