The ImageJ User Guide provides a detailed overview of ImageJ (and inherently Fiji), the standard in scientific image analysis (see XXVI Focus on Bioimage Informatics). It was thought as a comprehensive, fully-searchable, self-contained, annotatable manual (see Conventions Used in this Guide). A HTML version is also available as well as printer-friendly booklets (see Guide Formats). Its latest version can always be obtained from http://imagej.nih.gov/ij/docs/guide. The source files are available through a Git version control repository at http://fiji.sc/guide.git.

Given ImageJ’s heavy development this guide will always remain incomplete. All ImageJ users and developers are encouraged to contribute to the ImageJ documentation resources (see Getting Involved).

Contents

Release Notes for ImageJ 1.46r vii
Noteworthy viii
Macro Listings ix
Conventions x

I Getting Started

1 Introduction 1

2 Installing and Maintaining ImageJ 2
  2.1 ImageJDistributions 2
  2.2 Related Software 3
  2.3 ImageJ2 5

3 Getting Help 5
  3.1 Help on Image Analysis 5
  3.2 Help on ImageJ 6

II Working with ImageJ

4 Using Keyboard Shortcuts 8

5 Finding Commands 8

6 Undo and Redo 9

7 Image Types and Formats 10

8 Stacks, Virtual Stacks and Hyperstacks 12

9 Color Images 14

10 Selections 17
  10.1 Manipulating ROIs 18
  10.2 Composite Selections 19
  10.3 Selections With Sub-pixel Coordinates 19

11 Overlays 19

12 3D Volumes 21

13 Settings and Preferences 22

III Extending ImageJ

14 Macros 23
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>29.6 Noise</td>
<td>110</td>
</tr>
<tr>
<td>29.7 Shadows</td>
<td>111</td>
</tr>
<tr>
<td>29.8 Binary</td>
<td>112</td>
</tr>
<tr>
<td>29.9 Math</td>
<td>117</td>
</tr>
<tr>
<td>29.10 FFT</td>
<td>120</td>
</tr>
<tr>
<td>29.11 Filters</td>
<td>124</td>
</tr>
<tr>
<td>29.12 Batch</td>
<td>126</td>
</tr>
<tr>
<td>29.13 Image Calculator</td>
<td>128</td>
</tr>
<tr>
<td>29.14 Subtract Background</td>
<td>129</td>
</tr>
<tr>
<td>29.15 Repeat Command [R]</td>
<td>131</td>
</tr>
<tr>
<td>30 Analyze</td>
<td>132</td>
</tr>
<tr>
<td>30.1 Measure [m]</td>
<td>132</td>
</tr>
<tr>
<td>30.2 Analyze Particles</td>
<td>132</td>
</tr>
<tr>
<td>30.3 Summarize</td>
<td>135</td>
</tr>
<tr>
<td>30.4 Distribution</td>
<td>135</td>
</tr>
<tr>
<td>30.5 Label</td>
<td>136</td>
</tr>
<tr>
<td>30.6 Clear Results</td>
<td>136</td>
</tr>
<tr>
<td>30.7 Set Measurements</td>
<td>136</td>
</tr>
<tr>
<td>30.8 Set Scale</td>
<td>139</td>
</tr>
<tr>
<td>30.9 Calibrate</td>
<td>139</td>
</tr>
<tr>
<td>30.10 Histogram [h]</td>
<td>140</td>
</tr>
<tr>
<td>30.11 Plot Profile [k]</td>
<td>142</td>
</tr>
<tr>
<td>30.12 Surface Plot</td>
<td>143</td>
</tr>
<tr>
<td>30.13 Gels</td>
<td>144</td>
</tr>
<tr>
<td>30.14 Tools</td>
<td>146</td>
</tr>
<tr>
<td>31 Plugins</td>
<td>156</td>
</tr>
<tr>
<td>31.1 Macros</td>
<td>156</td>
</tr>
<tr>
<td>31.2 Shortcuts</td>
<td>157</td>
</tr>
<tr>
<td>31.3 Utilities</td>
<td>158</td>
</tr>
<tr>
<td>31.4 New</td>
<td>161</td>
</tr>
<tr>
<td>31.5 Compile and Run</td>
<td>162</td>
</tr>
<tr>
<td>32 Window</td>
<td>163</td>
</tr>
<tr>
<td>32.1 Show All [ ]</td>
<td>163</td>
</tr>
<tr>
<td>32.2 Put Behind [tab]</td>
<td>163</td>
</tr>
<tr>
<td>32.3 Cascade</td>
<td>163</td>
</tr>
<tr>
<td>32.4 Tile</td>
<td>163</td>
</tr>
<tr>
<td>33 Help</td>
<td>164</td>
</tr>
<tr>
<td>33.1 ImageJ Website</td>
<td>164</td>
</tr>
<tr>
<td>33.2 ImageJ News</td>
<td>164</td>
</tr>
<tr>
<td>33.3 Documentation</td>
<td>164</td>
</tr>
<tr>
<td>33.4 Installation</td>
<td>164</td>
</tr>
<tr>
<td>33.5 Mailing List</td>
<td>164</td>
</tr>
<tr>
<td>33.6 Dev. Resources</td>
<td>164</td>
</tr>
<tr>
<td>33.7 Plugins</td>
<td>164</td>
</tr>
<tr>
<td>33.8 Macros</td>
<td>164</td>
</tr>
<tr>
<td>33.9 Macro Functions</td>
<td>164</td>
</tr>
<tr>
<td>33.10 Update ImageJ</td>
<td>165</td>
</tr>
<tr>
<td>33.11 Refresh Menus</td>
<td>165</td>
</tr>
<tr>
<td>33.12 About Plugins</td>
<td>165</td>
</tr>
<tr>
<td>33.13 About ImageJ</td>
<td>165</td>
</tr>
</tbody>
</table>

**VI Keyboard Shortcuts**

**34 Key Modifiers**
- 34.1 Alt Key Modifications | 167
- 34.2 Shift Key Modifications | 168
- 34.3 Ctrl (or Cmd) Key Modifications | 169
- 34.4 Space Bar | 169
- 34.5 Arrow Keys | 169

**35 Toolbar Shortcuts**

- Credits | 171
- ImageJ Related Publications | 174
- List of Abbreviations and Acronyms | 181
- Index | 182
- Colophon | 186
Summarized Release Notes for ImageJ 1.46r

Selections with sub-pixel resolution .................................................. 19
Improved handling of Overlays .......................................................... 19
Fourteen new macro functions ......................................................... 24
New tools: Arrow, Brush, Developer Menu, Flood Filler, LUT Menu, Overlay Brush,
  Pencil, Pixel Inspector, Spray Can and Stacks Menu .......................... 38
Plugin Tools ..................................................................................... 41
Customizable crosshair cursor ......................................................... 47
Improved AVI support ...................................................................... 55
New command: File>Import>XY Coordinates ....................................... 55
Undo support extended to Edit>Selection>commands ............................. 63
New command: Edit>Selection>Interpolate ......................................... 64
Edit>Selection>Line to Area now works with one pixel wide straight lines . . 66
New command: Edit>Selection>Image to Selection ............................... 67
New command: Edit>Options>Reset .................................................. 74
Image>Color>Merge Channels extended to 7 channels ......................... 83
ColorChoosers display hex color values ........................................... 85
Improved Image>Stacks>Tools>Concatenate ....................................... 93
New command: Image>Transform>Bin ............................................... 99
New commands: Image>Transform>Image to Results and Results to Image .100
New command: Image>Overlay>Labels ............................................. 104
New command: Image>Overlay>Overlay Options ................................. 105
New command: Edit>Selection>Gaussian Blur 3D ............................... 125
Live histograms .............................................................................. 140
Live profile plots ............................................................................ 142
Much improved Curve Fitter ............................................................ 148
The ROI Manager now uses an overlay to display selections when in Show All mode ....................................................... 151
New command: ROI Manager’s Print List .......................................... 152
New command: Analyze>Tools>Synchronize Windows ....................... 154
New commands: Plugins>New>Macro/Plugin Tool ............................. 161

Noteworthy

I Frontmost Window and Window Activation ..................................... 9
II Image Types: Lossy Compression and Metadata ............................... 12
III Opening Virtual Stacks by Drag & Drop ........................................ 13
IV Replacing Red with Magenta in RGB Images .................................. 16
V Toggling Calibrated Units ............................................................. 29
VI Opening File Paths in the Log Window .......................................... 46
VII Organizing Commands in the Menu Bar ..................................... 48
VIII Opening Files: File>Open..., File>Import> and Drag & Drop .......... 50
IX Reducing Memory Requirements When Importing Images ............... 53
X Warning on JPEG Compression ................................................... 57
XI Drawing Lines Wider Than One–Pixel .......................................... 62
XII Transferring Selections Between Images ..................................... 64
XIII Converting Composite Selections ............................................... 66
XIV Applying Auto Brightness/Contrast to Entire Stacks ...................... 76
XV Display Range of DICOM Images ............................................... 78
XVI Brightness/Contrast of High Bit–Depth Images ............................. 79
XVII Embedding Color Annotations in Grayscale Images .................... 86
XVIII Working with Zoomed Canvases ............................................. 101
XIX Hexdecimal Color Values .......................................................... 103
XX Creating Binary Masks ................................................................ 113
XXI Skeletonize vs Skeletonize 3D .................................................... 115
XXII Interpreting Binary Images ....................................................... 118
XXIII Global Calibrations .................................................................. 141
XXIV Using Scanners in Densitometry .............................................. 144
XXV Selecting ROIs in the ROI Manager ........................................... 153
XXVI Focus on Bioimage Informatics ............................................... 173

Please note that this is not an extensive list. Detailed release notes for version 1.46r are available on the ImageJ news web site: http://imagej.nih.gov/ij/notes.html.
Part I
Getting Started

This part provides basic information on ImageJ installation, troubleshooting and update strategies. It discusses Fiji and ImageJ2 as well as third-party software related to ImageJ. Being impossible to document all the capabilities of ImageJ without exploring technical aspects of image processing, external resources allowing willing readers to know more about digital signal processing are also provided.

1 Introduction

ImageJ is a public domain Java image processing and analysis program inspired by NIH Image for the Macintosh. It runs, either as an online applet or as a downloadable application, on any computer with a Java 1.5 or later virtual machine. Downloadable distributions are available for Windows, Mac OS X and Linux. It can display, edit, analyze, process, save and print 8-bit, 16-bit and 32-bit images. It can read many image formats including TIFF, GIF, JPEG, BMP, DICOM, FTTS and ‘raw’. It supports ‘stacks’ (and hyperstacks), a series of images that share a single window. It is multithreaded, so time-consuming operations such as image file reading can be performed in parallel with other operations.

It can calculate area and pixel value statistics of user-defined selections. It can measure distances and angles. It can create density histograms and line profile plots. It supports standard image processing functions such as contrast manipulation, sharpening, smoothing, edge detection and median filtering.

It does geometric transformations such as scaling, rotation and flips. Image can be zoomed up to 32:1 and down to 1:32. All analysis and processing functions are available at any magnification factor. The program supports any number of windows (images) simultaneously, limited only by available memory.

Spatial calibration is available to provide real world dimensional measurements in units such as millimeters. Density or gray scale calibration is also available.

ImageJ was designed with an open architecture that provides extensibility via Java plugins. Custom acquisition, analysis and processing plugins can be developed using ImageJ’s built in editor and Java compiler. User-written plugins make it possible to solve almost any image processing or analysis problem.

Being public domain open source software, an ImageJ user has the four essential freedoms defined by the Richard Stallman in 1986: 1) The freedom to run the program, for any purpose; 2) The freedom to study how the program works, and change it to make it do what you wish; 3) The freedom to redistribute copies so you can help your neighbor; 4) The freedom to improve the program, and release your improvements to the public, so that the whole community benefits.

ImageJ is being developed on Mac OSX using its built in editor and Java compiler, plus the BBEdit editor and the Ant build tool. The source code is freely available. The author, Wayne Rasband (wsr@nih.gov), is a Special Volunteer at the National Institute of Mental Health, Bethesda, Maryland, USA.

See also: History of ImageJ at imagejdev.org

2 Installing and Maintaining ImageJ


The downloaded package may not contain the latest bug fixes so it is recommended to upgrade ImageJ right after a first installation. Updating IJ consists only of running Help> Update ImageJ... which will install the latest ij.jar in the ImageJ folder (on Linux and Windows) or inside the ImageJ.app (on Mac OSX).

Help> Update ImageJ... can be used to upgrade (or downgrade) the ij.jar file to release updates or daily builds. Release updates are announced frequently on the IJ news website and are labelled alphabetically (e.g., v.1.43m). Typically, these releases contain several new features and bug fixes, described in detail on the ImageJ News page. Daily builds, on the other hand, are labelled with numeric sub-indexes (e.g., v.1.43n4) and are often released without documentation. Nevertheless, if available, release notes for daily builds can be found at http://imagej.nih.gov/ij/source/release-notes.html. When a release cycle ends (v.1.42 ended with 1.42q, v.1.43 with 1.43u, etc.) an installation package is created, downloadable from http://imagej.nih.gov/ij/download.html. Typically, this package is bundled with a small list of add-ons (Macros, Scripts and Plugins).

See also: Luts, Macros and Tools Updater, a macro toolset that performs live-updating of macros listed on the ImageJ web site

2.1 ImageJDistributions

ImageJ alone is not that powerful: it’s real strength is the vast repertoire of Plugins that extend ImageJ’s functionality beyond its basic core. The many hundreds, probably thousands, freely available plugins from contributors around the world play a pivotal role in ImageJ’s success [2][3]. Running Help> Update ImageJ..., however, will not update any of the plugins you may have installed.

ImageJ add-ons (Plugins, Scripts and Macros) are available from several sources (ImageJ’s plugins page [Help> Plugins...], ImageJ Information and Documentation Portal and Fiji’s webpage, among others) making manual updates of a daunting task. This reason alone, makes it extremely convenient the use of ImageJdistributions bundled with a pre-organized collection of add-ons.

Below is a list of the most relevant projects that address the seeming difficult task of organizing and maintaining ImageJ beyond its basics. If you are a life scientist and have doubts about which distribution to choose you should opt for Fiji. It is heavily maintained, offers an automatic updater, improved scripting capabilities and ships with powerful plugins. More specialized adaptations of ImageJ are discussed in Software Packages Built on Top of ImageJ.

Fiji

Fiji (Fiji Is Just ImageJ—Batteries included) is a distribution of ImageJ together with Java, Java 3D and several plugins organized into a coherent menu structure. Citing its developers,
“Fiji compares to ImageJ as Ubuntu compares to Linux”. The main focus of Fiji is to assist research in life sciences, targeting image registration, stitching, segmentation, feature extraction and 3D visualization, among others. It also supports many scripting languages (BeanScript, Clojure, Python, Ruby, see Scripting in Other Languages). Importantly, Fiji ships with a convenient updater that knows whether your files are up-to-date, obsolete or locally modified. Comprehensive documentation is available for most of its plugins. The Fiji project was presented publicly for the first time at the ImageJ User and Developer Conference in November 2008.

MBF ImageJ

The MBF ImageJ bundle or ImageJ for Microscopy (formerly WCIF-ImageJ) features a collection of plugins and macros, collated and organized by Tony Collins at the MacBiophotonics facility, McMaster University. It is accompanied by a comprehensive manual describing how to use the bundle with light microscope data. It is a great resource for microscopists but is not maintained actively, lagging behind the development of core ImageJ.

Note that you can add plugins from MBF ImageJ to Fiji, combining the best of both programs. Actually, you can use multiple ImageJ distributions simultaneously, assemble your own ImageJ bundle by gathering the plugins that best serve your needs (probably, someone else at your institution already started one?) or create symbolic links to share plugins between different installations.

See also: Description of all ImageJ related projects at ImageDev

2.2 Related Software

2.2.1 Software Packages Built on Top of ImageJ

Bio7 Bio7 is an integrated development environment for ecological modeling with a main focus on individual based modeling and spatially explicit models. Bio7 features: Statistical analysis (using R); Spatial statistics; Fast communication between R and Java; BeanShell and Groovy support; Sensitivity analysis with an embedded flowchart editor and creation of 3D OpenGL (Jogl) models (see also RImageJ in ImageJ Interoperability).

BoneJ BoneJ is a collection of tools for trabecular geometry and whole bone shape analysis.

µManager Micro-Manager is a software package for control of automated microscopes. It lets you execute common microscope image acquisition strategies such as time-lapses, multi-channel imaging, z-stacks, and combinations thereof. µManager works with microscopes from all four major manufacturers, most scientific-grade cameras and many peripherals used in microscope imaging.

MRI–CIA MRI Cell Image Analyzer, developed by the Montpellier RIO Imaging facility (CNRS), is a rapid image analysis application development framework, adding visual scripting interface to ImageJ’s capabilities. It can create batch applications as well as interactive applications. The applications include the topics—DNA combing”, “quantification of stained proteins in cells”, “comparison of intensity ratios between nuclei and cytoplasm” and “counting nuclei stained in different channels”.

ObjectJ ObjectJ, the successor of object-image, supports graphical vector objects that non-destructively mark images on a transparent layer. Vector objects can be placed manually or by macro commands. Composite objects can encapsulate different color-coded marker structures in order to bundle features that belong together. ObjectJ supports back-and-forth navigation between results and images. The results table supports statistics, sorting, color coding, qualifying and macro access.

SalsaJ SalsaJ is a student-friendly software developed specifically for the EU-HOU project. It is dedicated to image handling and analysis of astronomical images in the classroom. SalsaJ has been translated into several languages.

TrakEM2 TrakEM2 is a program for morphological data mining, three-dimensional modeling and image stitching, registration, editing and annotation [13]. TrakEM2 is distributed with Fiji and capable of:

- **3D modeling** Objects in 3D, defined by sequences of contours, or profiles, from which a skin, or mesh, can be constructed, and visualized in 3D.
- **Relational modeling** The extraction of the map that describes links between objects. For example, which neuron contacts which other neurons through which many and which synapses.

See also: BioImageXD, Endrov, Image SXM

2.2.2 ImageJ Interoperability

Several packages exist that allow ImageJ to interact with other applications/environments:

- **Bitplane Imaris** ImarisXT can load and execute ImageJ plugins. bpidmarisAdapter (Windows only and requiring valid licenses for Imaris and ImarisXT) allows the exchange of images between Imaris and ImageJ.
- **CellProfiler** CellProfiler [17] features RunImageJ, a module that allows ImageJ plugins to be run in a CellProfiler pipeline.
- **Icy** Icy, an open source community software for bio-imaging, executes ImageJ plugins with almost 100% plugin compatibility.
- **Knime** Knime (Konstanz Information Miner) contains several image processing nodes (KNIP) that are capable of executing ImageJ plugins and macros.

Open Microscopy Environment All Open Microscopy Environment projects such as Bio-Formats, VisBio and OME integrate well with ImageJ.

RImageJ — R bindings for ImageJ Bindings between ImageJ and R (GNU S) — The free software environment for statistical computing and graphics. The documentation for RImageJ is available at http://cran.r-project.org/web/packages/RImageJ/RImageJ.pdf (see also Bio7 in Software Packages Built on Top of ImageJ).

MIJ — Matlab–ImageJ bi-directional communication A Java package for bi-directional data exchange between Matlab and ImageJ, allowing to exchange images between the two imaging software. MIJ also allows MATLAB to access all built-in functions of ImageJ as well as third-party ImageJ plugins. The developers provide more information on the MIJ and Matlab File Exchange websites. Fiji features MIJ, which makes even more convenient to use the libraries and functions provided by Fiji’s components from within Matlab.

See also: ImageJ related links, list of related imaging software on the ImageJ2 website
2.3 ImageJ2

ImageJDev is a federally funded, multi-institution project dedicated to the development of the next-generation version of ImageJ: “ImageJ2”. ImageJ2 is a complete rewrite of ImageJ, that includes the current, stable version ImageJ (“ImageJ1”) with a compatibility layer so that old-style plugins and macros can run the same as they currently do in ImageJ1. Below is a summary of the ImageJDev project aims:

- To create the next generation version of ImageJ and improve its core architecture based on the needs of the community.
- To ensure ImageJ remains useful and relevant to the broadest possible community, maintaining backwards compatibility with ImageJ1 as close to 100% as possible.
- Expand functionality by interfacing ImageJ with existing open-source programs.
- To lead ImageJ development with a clear vision, avoiding duplication of efforts.
- To provide a central online resource for ImageJ: program downloads, a plugin repository, developer resources and more.

Be sure to follow the ImageJ2 project news and the ImageDev blog for updates on this exciting project.

3 Getting Help

3.1 Help on ImageJ

Below is a list of online resources (in no particular order) related to image processing and scientific image analysis, complementing the list of external resources on the IJ web site.

Ethics in Scientific Image Processing

- Online learning Tool for Research Integrity and Image Processing
  This website, created by the Office of Research Integrity, explains what is appropriate in image processing and science and what is not.
- Digital Imaging: Ethics (at the Cellular Imaging Facility Core, SEHSC)
  This website, compiled by Douglas Cromeey at the University of Alabama – Birmingham, discusses thoroughly the topic of digital imaging ethics. It is recommended for all scientists. The website contains links to several external resources, including:

Scientific Image Processing

- What you need to know about scientific image processing
  Simple and clear, this Fiji webpage explains basic aspects of scientific image processing.

See also: ImageJ Related Publications on page 174

3.2 Help on ImageJ

Below is a list of the ImageJ help resources that complement this guide (see Guide Formats).

1. The ImageJ online documentation pages
   Can be accessed via the Help→Documentation… command.
2. The Fiji webpage:
   http://fiji.sc/
3. The ImageJ Information and Documentation Portal (ImageJ wikipage):
   http://imagejdocu.tudor.lu/doku.php
4. Video tutorials on the ImageJ Documentation Portal and the Fiji YouTube channel:
5. The ImageJ for Microscopy manual
   http://www.machiphotonics.ca/imagej/
6. Several online documents, most of them listed at:
7. Mailing lists:
   (a) ImageJ — http://imagej.nih.gov/ij/list.html
      General user and developer discussion about ImageJ. Can be accessed via the Help→Mailing List… command. This list is also mirrored at Nabble and Gmane. You may find it easier to search and browse the list archives on these mirrors. Specially useful are the RSS feeds and the frames and threads view provided by Gmane.

http://imagej.nih.gov/ij/docs/examples/
Part II

Working with ImageJ

This part introduces some basic aspects of ImageJ so that you can use the software more efficiently. It also introduces some important terms and concepts used throughout this guide. You may skip it if you already use the program efficiently and are familiar with terms such as Virtual Stacks, Hyperstacks, Pseudocolor Images, Color Composite Images or Composite Selections.

4 Using Keyboard Shortcuts

You’ll learn more and more shortcut keys as you use ImageJ, because (almost) all shortcuts are listed throughout ImageJ menus. Similarly, in this guide each command has its shortcut listed on its name (flanked by square brackets). Please note that the notation for these key-bindings is case sensitive, i.e., Shift-modifiers are not explicitly mentioned (a capital A means Shift-A) and assumes that Require control key for shortcuts in Edit→Options→Misc… is unchecked (i.e., except when using the IJ Editor or the Text Tool, you won’t have to hold down the Control key to use menu shortcuts). For example, the command Edit→Invert [I] can be evoked by Shift-I or Ctrl-Shift-I if Require control key for shortcuts is checked. The full list of ImageJ shortcuts can be retrieved at any time using the Plugins→Utilities→List Shortcuts… command.

There are three modifier keys in ImageJ:

Control (Command Key on Apple keyboards) Denoted by ‘Ctrl’ or [Ctrl] in this document. Although a control key is typically present on Apple keyboards, on a Macintosh computer running ImageJ the Command key [X-Command] replaces the functionality of the Control key of other operating systems. For sake of simplification, ‘Ctrl’ will always refer to both throughout this guide.

Shift Denoted by ‘Shift’ or [Shift] in this document.

Alt Denoted by ‘Alt’ or [Alt] in this document. This is also the ‘Option’ or ‘Meta’ key on many keyboards. In ImageJ, it is also used to type special unit symbols such as µ (Alt-M) or Å (Alt-Shift-A).

See also: Keyboard Shortcuts, Plugins→Shortcuts

5 Finding Commands

Navigating through the extensive list of ImageJ commands, macros and plugins may be quite cumbersome. Through its built-in Command Finder/Launcher[8], ImageJ offers an expedite alternative that allows you to retrieve commands extremely fast: Plugins→Utilities→Find Commands… [9].

In addition, ImageJ features a find function that locates macros, scripts and plugins source (.java) files on your computer: the Plugins→Utilities→Search… command. Because most of IJ source files contain circumstanced comments, you can use this utility to retrieve files
I Frontmost Window and Window Activation

In ImageJ, all operations are performed on the active (frontmost) image (which has its title bar highlighted). If a window is already open it will activate when its opening command is re-run, e.g., if the B&C window is already opened (Image>Adjust>Brightness/Contrast... [C]), pressing its keyboard shortcut (Shift+C) will activate it.

Pressing Enter on any image will bring the Main ImageJ window to the foreground. In addition, it is also possible to permanently place the main window above all other windows (see Floating Behavior of Main Window).

related not only to a image processing routine (e.g., background or co-localization) but also to a practical context such as radiogram, cell or histology. Indeed, ImageJ source files contain detailed annotations useful to both developers and regular users that want to know more about ImageJ routines and algorithms.

Search... and Find Commands... [I] are described in detail in Plugins>Utilities>.

See also: Control Panel... [U], Keyboard Shortcuts and SourceCodeRetriever, a macro that searches for a menu entry and retrieves the source file of the respective command

6 Undo and Redo

Probably the first thing you will notice is that ImageJ does not have a large undo/redo buffer. Undo (Edit>Undo [z]) is currently limited to the most recent image editing/filtering operation. With time you will appreciate that this is necessary to minimize memory overhead. Nevertheless, with IJ 1.45 and later, Undo [z] is, in most cases, undoable and can be applied to multiple images if Keep multiple undo buffers is checked in Edit>Options>Memory & Threads.

If you cannot recover from a mistake, you can always File>Revert [r] to reset the image to its last saved state. For selections, Edit>Selection>Restore Selection [E] can be used to recover any misdealt selection.

In ImageJ the equivalent to ‘Redo’ is the Process>Repeat Command [R], that re-runs the previous used command (skipping Edit>Undo [z] and File>Open... [o] commands).

See also: Plugins>Utilities>Reset... Multi Undo plugin

7 Image Types and Formats

Digital Images are two-dimensional grids of pixel intensities values with the width and height of the image being defined by the number of pixels in x (rows) and y (columns) direction. Thus, pixels (picture elements) are the smallest single components of images, holding numeric values – pixel intensities – that range between black and white. The characteristics of this range, i.e., the number of unique intensity (brightness) values that can exist in the image is defined as the bit-depth of the image and specifies the level of precision in which intensities are coded, e.g.: A 2-bit image has $2^2 = 4$ tones: 00 (black), 01 (gray), 10 (gray), and 11 (white). A 4-bit image has $2^4 = 16$ tones ranging from 0000 (0) to 1111 (16), etc. In terms of bits per pixel (bpp), the most frequent types of images (Image>Types...) that ImageJ deals with are (ImageJ2 supports many more types of image data):

8–bit Images that can display 256 ($2^8$) gray levels (integers only).
16–bit Images that can display 65,536 ($2^{16}$) gray levels (integers only).
32–bit Images that can display 4,294,967,296 ($2^{32}$) gray levels (real numbers). In 32-bit images, pixels are described by floating point values and can have ANY intensity value including NaN (Not a Number).

RGB Color Color Images that can display 256 values in the Red, Green and Blue channel. These are 24–bit ($2^{3×8}$ images). RGB color images can also be 32-bit color images (24-bit color images with additional eight bits coding alpha blending values, i.e., transparency).

Native Formats

Natively (i.e. without the need of third-party plugins) ImageJ opens the following formats: TIFF, GIF, JPEG, PNG, DICOM, BMP, PGM and FITS. Many more formats are supported with the aid of plugins. These are discussed in Non-native Formats.

TIFF (Tagged Image File Format) is the ‘default’ format of ImageJ (cf. File>Save [s]). Images can be 1-bit, 8-bit, 16-bit (unsigned), 32-bit (real) or RGB color. TIFF files with multiple images of the same type and size open as Stacks or Hyperstacks. ImageJ opens lossless compressed TIFF files (see II Image Types: Lossy Compression and Metadata) by the LZW, PackBits and ZIP (Deflate/Inflate) [2] compression schemes. In addition, TIFF files can be opened and saved as ZIP archives. Tags and information needed to import the file (number of images, offset to first images, gap between images) are printed to the Log Window when ImageJ is running in Debug Mode (Edit>Options>Misc... see Settings and Preferences).

1 A numeric variable is signed if it can represent both positive and negative numbers, and unsigned if it can only represent positive numbers.
### Image Types and Formats

DICOM (Digital Imaging and Communications in Medicine) is a standard popular in the medical imaging community. Support in ImageJ is limited to uncompressed DICOM files, DICOM files containing multiple images open as Stacks. Use `Image > Show Info...` to display the DICOM header information. A DICOM sequence can be opened using `File > Import > Image Sequence...` or by dragging and dropping the folder on the `ImageJ` window. Imported sequences are sorted by image number instead of filename and the tags are preserved when DICOM images are saved in TIFF format. ImageJ supports custom DICOM dictionaries, such as the one at [http://imagej.nih.gov/ij/download/docs/DICOM_Dictionary.txt](http://imagej.nih.gov/ij/download/docs/DICOM_Dictionary.txt). More information can be found at the Center for Advanced Brain Imaging.

FITS (Flexible Image Transport System) image is the format adopted by the astronomical community for data interchange and archival storage. Use `Image > Show Info...` to display the FITS header. More information here.

PGM (Portable GrayMap), PBM (Portable BitMap) and PPM (Portable PixMap) are simple image formats that use an ASCII header. More information here.

AVI (Audio Video Interleave) is a container format which can contain data encoded in many different ways. ImageJ only supports uncompressed AVIs, various YUV 4:2:2 compressed formats, and PNG or JPEG-encoded individual frames. Note that most MJPG (motion-JPEG) formats are not read correctly. Attempts to open AVIs in other formats will fail.

**See also:** Non-native Formats, II Image Types: Lossy Compression and Metadata, X Warning on JPEG Compression

### Non-native Formats

When opening a file, ImageJ first checks whether it can natively handle the format. If ImageJ does not recognize the type of file it calls for the appropriate reader plugin using `HandleExtraFileTypes`, a plugin bundled with ImageJ. If that fails, it tries to open the file using the OME Bio-Formats library (if present), a remarkable plugin that supports more than one hundred of the most common file formats used in microscopy. If nevertheless the file cannot be opened, an error message is displayed.

Because both these plugins are under active development, it is important that you keep them updated. The OME Bio-Formats library can be updated using its self-updating plugin (Plugins > LOCI > Update LOCI Plugin...) or Fiji’s built-in updater (Help > Update Fiji...). The following websites provide more information on the OME Bio-Formats:

- [http://loci.wisc.edu/bio-formats/imagej](http://loci.wisc.edu/bio-formats/imagej)
- [http://fiji.sc/Bio-Formats](http://fiji.sc/Bio-Formats)
- [http://loci.wisc.edu/bio-formats/using-bio-formats](http://loci.wisc.edu/bio-formats/using-bio-formats)

In addition, the ImageJ web site lists more than sixty plugins that recognize more ‘exotic’ file formats. The ImageJ Documentation Portal also maintains a (somewhat outdated) list of file formats that are supported by ImageJ.

**See also:** Native Formats, File > Import..., II Image Types: Lossy Compression and Metadata, X Warning on JPEG Compression, Acquisition plugins, Input/Output plugins

### II Image Types: Lossy Compression and Metadata

Two critical aspects to keep in mind when converting images:

**Lossy compression** Transcoding an image into a format that uses lossy compression will alter the original data, introducing artifacts (see X Warning on JPEG Compression). This is the case, e.g., for JPEG formats (with the exception of some JPEG2000 images that use lossless compression). As such, these types of data are intended for human interpretation only and are not suitable for quantitative analyses.

**Metadata** In ImageJ, metadata associated with the image, such as scale, gray value calibration and user comments is only supported in tiff and zip (compressed tiff) images. In addition, selections and Overlays are also saved in the TIFF header (cf. File > Save [s]). None of the above is saved in other formats (cf. Native Formats).

### 8 Stacks, Virtual Stacks and Hyperstacks

#### Stacks

ImageJ can display multiple spatially or temporally related images in a single window. These image sets are called stacks. The images that make up a stack are called slices. In stacks, a pixel (which represents 2D image data in a bitmap image) becomes a voxel (volumetric pixel), i.e., an intensity value on a regular grid in a three dimensional space.

All the slices in a stack must be the same size and bit depth. A scrollbar provides the ability to move through the slices and the slider is preceded by a play/pause icon that can be used to start/stop stack animation. Right-clicking on this icon runs the Animation Options... [Alt/] dialog box.

Most ImageJ filters will, as an option, process all the slices in a stack. ImageJ opens multi-image TIFF files as a stack, and saves stacks as multi-image TIFFs. The `File > Import > Raw...` command opens other multi-image, uncompressed files. A folder of images can be opened as a stack either by dragging and dropping the folder onto the `ImageJ` window or by choosing `File > Import > Image Sequence...` To create a new stack, simply choose `File > New > Image...` [n] and set the `Slices` field to a value greater than one. The `Image > Stacks` submenu contains commands for common stack operations.

**See also:** Stacks Menu, Stack Manipulations on Fiji website, Image5D

#### Virtual Stacks

Virtual stacks are disk resident (as opposed to RAM resident) and are the only way to load image sequences that do not fit in RAM. There are several things to keep in mind when working with virtual stacks:

- Virtual stacks are read-only, so changes made to the pixel data are not saved when you switch to a different slice. You can work around this by using macros (e.g., Process Virtual Stack) or the `Process > Batch > Virtual Stack...` command
- You can easily run out of memory using commands like `Image > Crop` [X] because any stack generated from commands that do not generate virtual stacks will be RAM resident.

### Virtual Stacks

Virtual stacks are read-only, so changes made to the pixel data are not saved when you switch to a different slice. You can work around this by using macros (e.g., Process Virtual Stack) or the `Process > Batch > Virtual Stack...` command.

You can easily run out of memory using commands like `Image > Crop` [X] because any stack generated from commands that do not generate virtual stacks will be RAM resident.
Color Images

ImageJ deals with color mainly in three ways: pseudocolor images, RGB images, RGB/HSB stacks, and composite images.

Pseudocolor Images

A pseudocolor (or indexed color) image is a single channel gray image (8, 16 or 32-bit) that has color assigned to it via a lookup table or LUT. A LUT is literally a predefined table of gray values with matching red, green and blue values so that shadows of gray are displayed as colorized pixels. Thus, differences in color in the pseudo-colored image reflect differences in intensity of the object rather than differences in color of the specimen that has been imaged.

8-bit indexed color images (such as GIFs) are a special case of pseudocolor images as their lookup table is stored in the file with the image. These images are limited to 256 colors (24-bit RGB images allow 16.7 million of colors, see Image Types and Formats) and concomitantly smaller file sizes. Reduction of true color values to a 256 color palette is performed by color quantization algorithms. ImageJ uses the Heckbert’s median-cut color quantization algorithm (see Image Types menu), which, in most cases, allows indexed color images to look nearly identical to their 24-bit originals.

SEE ALSO: Image > Lookup Tables > and LUT Menu

True Color Images

As described in Image Types and Formats, true color images such as RGB images reflect genuine colors, i.e., the green in an RGB image reflects green color in the specimen. Color images are typically produced by color CCD cameras, in which color filter arrays (Bayer masks) are placed over the image sensor.

Color Spaces and Color Separation

Color spaces describe the gamut of colors that image-handling devices deal with. Because human vision is trichromatic, most color models represent colors by three values. Mathematically, these values (color components) form a three-dimensional space such as the RGB, HSB, CIE Lab or YUV color space.

RGB (Red, Green, Blue) is the most commonly-used color space. However, other alternatives such as HSB (Hue, Saturation, Brightness) provide significant advantages when processing color information. In the HSB color space, Hue describes the attribute of pure color, and therefore distinguishes between colors. Saturation (sometimes called “purity” or “vibrancy”) characterizes the shade of color, i.e., how much white is added to the pure color. Brightness (also know as Value – HSV system) describes the overall brightness of the color (see also, the color palette of Color Picker window). In terms of digital imaging processing, using the HSB system over the traditional RGB is often advantageous, e.g., since the Brightness component of an HSB image corresponds to the grayscale version of that image, processing only the brightness channel in

This page is partially extracted from the MBF ImageJ online manual at http://www.machiophotonics.ca/imagej/colour_image_procesi.htm.
Conveying Color Information

People see color with significant variations. Indeed, the popular phrase “One picture is worth ten thousand words” may not apply to certain color images, specially those that do not follow the basic principles of Color Universal Design. Citing Masataka Okabe and Kei Ito:

Colorblind people can recognize a wide ranges of colors. But certain ranges of colors are hard to distinguish. The frequency of colorblindness is fairly high. One in 12 Caucasian (8%), one in 20 Asian (5%), and one in 25 African (4%) males are so-called ‘red–green’ colorblind.

There are always colorblind people among the audience and readers. There should be more than ten colorblind in a room with 250 people (assuming 50% male and 50% female).

There is a good chance that the paper you submit may go to colorblind reviewers. Supposing that your paper will be reviewed by three white males (which is not unlikely considering the current population in science), the probability that at least one of them is colorblind is whopping 22%!

One practical point defined by the Color Universal Design is the use of magenta in red–green overlays (see also [66]). Magenta is the equal mixture of red and blue. Colorblind people that have difficulties recognizing the red component can easily recognize the blue hue. The region of double positive becomes white, which is easily distinguishable for colorblind. In ImageJ this is easily accomplished using the Image→Color→Merge Channels… command [20]. Segregation of color components (specially useful for quantification of histochemical staining) is also possible using Gabriel Landini’s Colour Deconvolution plugin. In addition, several other plugins related to color processing can be obtained from the ImageJ website.

Color Composite Images

In a composite image colors are handled through channels. The advantages with this type of image over plain RGB images are:

16 Last updated: 2012/10/02
1. Each channel is kept separate from the others and can be turned on and off using the ‘Channels’ tool (Image ➤ Channels Tool ... [2]). This feature allows, e.g., to perform measurements on a specific channel while visualizing multiple.

2. Channels can be 8, 16 or 32-bit and can be displayed with any lookup table.

3. More than 3 channels can be merged or kept separate.

10.1 Manipulating ROIs

Most of commands that can be useful in defining or drawing selections are available in the Edit ➤ Selection ➤ submenu and summarized in ROI manipulations. Listed below are the most frequent manipulations involving selections:

- **Adjusting** Area selections can be adjusted with the Brush Selection Tool. In addition, vertices of selections created with the Polygon Selection Tool and Segmented Line Selection Tool can be adjusted by Alt/Shift-clicking.

- **Deleting** Choose any of the selection tools and click outside the selection, or use Edit ➤ Selection ➤ Select None [A]. Use Edit ➤ Selection ➤ Restore Selection [E] to restore a deleted selection. With Overlays, an activated ROI can be deleted by pressing the Backspace [Delete] on Mac key.

- **Managing** A selection can be transferred from one image window to another by activating the destination window and running Edit ➤ Selection ➤ Restore Selection [E]. Alternatively, Analyze ➤ Tools ➤ Synchronize Windows to create ROIs across multiple images. Multiple selections can be stored as Overlays or in the ROI Manager list (Analyze ➤ Tools ➤ ROI Manager...).

- **Moving** Selections can be moved by clicking and dragging as long as the cursor is within the selection and has changed to an arrow. The status bar displays the coordinates of the upper left corner of the selection (or the bounding rectangle for non-rectangular selections) as it is being moved. To move the contents of a selection, rather than the selection itself, Edit ➤ Copy [C], Edit ➤ Paste [V], and then click within the selection and drag.

- **Nudging** Selections can be ‘nudged’ one pixel at a time in any direction using the arrow keys. Note that the up and down keys zoom the image in and out in the absence of selections (see Arrow Keys shortcuts).

- **Resizing** The Brush Selection Tool can be used to perform fine adjustments of ROI contours. Most ROIs can be resized one pixel at a time by holding Alt while using the arrow keys. In general (see Area Selection Tools and Line Selection Tools for details), selections are resized by dragging one of the selection handles. While dragging, holding Shift resizes the selection around its center, holding Alt imposes a fixed aspect ratio and holding Shift forces a 1:1 aspect ratio.

**See also:** Key Modifiers
10.2 Composite Selections

Composite selections are non-contiguous ROIs containing more than one cluster of pixels and/or ROIs containing internal holes. Composite ROIs are typically originated with the Brush Selection Tool but they can be defined with any other selection tool using key modifiers.

The following modifier keys can be used to create composite selections:

- **Shift** Drawing outside current selection while pressing Shift creates new content. To add a non-square rectangle or ellipse, the Shift key must be released after adding the selection
- **Alt** Drawing inside current selection while pressing Alt creates a hole removing content from the ROI

Note that some operations may not be performed properly on complex ROIs. In these cases, it may be useful to convert a composite ROI into a polygon using the **Edit > Selection > Enlarge** command as explained in XIII Converting Composite Selections.

**SEE ALSO:** Wand Tool, ROI2PolylineROI macro

10.3 Selections With Sub-pixel Coordinates

Since ImageJ 1.46, selections can be defined with subpixel accuracy, beyond the nominal pixel resolution of the image: Floating point selections. Line selections (see Line Selection Tools) are created with floating-point coordinates if the **Sub-pixel resolution** checkbox is active in **Edit > Options > Profile Plot Options**... Sub-pixel coordinates of pre-existing selections can be interpolated using the **Edit > Selection > Interpolate** command. Interpolated points are easily noticeable on small selections created on images zoomed 1200% or greater.

Interpolated selections. ROIs drawn with (left) or without (middle) sub-pixel accuracy. For line selections (see Line Selection Tools), this option can be enabled in **Edit > Options > Profile Plot Options**... by activating the **Sub-pixel resolution** checkbox. Pixel coordinates of area selections (see Area Selection Tools), can be interpolated using **Edit > Selection > Interpolate**. The image on the right is the output of `SubPixelSelections.js`, a script that demonstrates how to create selections at sub-pixel resolution without the need of setting any option in ImageJ.

**SEE ALSO:** Zoom, Magnifying Glass

11 Overlays

Overlays are non-active selections displayed ‘over’ the pixel data, on the image overlay, and are the core of non-destructive image processing in ImageJ. In a way you can think of the image overlay as an invisible ROI Manager in which selections are being added, allowing ROIs to be on ‘hold’. This concept of multiple distinct selections has been dramatically improved in ImageJ2 so we urge you to download IJ2 if multiple ROIs are important in your workflows.

Importantly, overlay selections are vector graphics composed of mathematically-defined paths (as opposed to raster graphics in which objects are defined by pixels) and are not affected by scaling, i.e., do not become pixelated. Most of overlay-related commands are listed in the **Image > Overlay**... and in the ROI Manager window (**Analyze > Tools > ROI Manager...**). Appearance of overlay selections can be adjusted using **Image > Overlay > Overlay Options**.../Labels...

As mentioned in II Image Types: Lossy Compression and Metadata, overlays are saved in the header of tif images, and do not need to be saved externally when using TIFF, the default file format of ImageJ. The major advantages of overlays are summarized below:

**Storage of ROIs** In ImageJ it is only possible to have a single ROI at a time. However, it is possible to add selections to the image overlay using **Image > Overlay > Add Selection...** [b]. Once added to the image overlay, ROIs can be re-activated by Alt-clicking, Control-clicking or long-pressing (1/4 second or longer). Activated ROIs can be deleted by pressing the **[Backspace]** key. Selections can also be added and recovered in bulk, using the **Image > Overlay > From ROI Manager/To ROI Manager commands**.

**Non-destructive annotations** Overlays are the best way of annotating images in ImageJ (examples). As vector graphics, overlays do not change pixel values, can be scaled without loss of quality even at high zoom levels (see XVIII Working with Zoomed Canvases) and can be displayed at different opacity values (see XIX Hexadecimal Color Values). RGB snapshots of the image with embedded overlays can be created by holding **Shift+F**, the shortcut for **Image > Overlay > Flatten [F]**. ‘Flattened’ images with the overlay rendered as pixel data are also created when saving the image as PNG or JPEG (**File > Save As...**), or when printing the image canvas (**File > Print...** [p]). The **Flatten** command is also listed in the ROI Manager.

**Image ROIs** An imageROI (image selection) is a ROI that displays an image as an overlay. As described in **Edit > Selection > Image to Selection...** and **Image > Overlay > Add Image...**, this allows multiple images to be blended on a single image canvas.
12 3D Volumes

Currently, the support for 3D ROIs (selections containing contiguous cluster of voxels) is somewhat limited in ImageJ. This limitation has been addressed by ImageJ2 and several IJ1 plugins. The list below summarizes some of the ImageJ plugins that deal effectively with multi-dimensional objects. Note that a manual installation of these tools as standalone ImageJ plugins is a challenging task given their special dependencies, reason why they are all bundled as part of Fiji.

3D Filters

Specialized 3D filters such as Process>Filters>Gaussian Blur 3D… can be installed to perform 3D operations. Examples are the 3D processing package by Thomas Bondier [51] and the 3D binary filters by Benjamin Schmid.

3D Object Counter

3D Object Counter (3D-OC) counts and qualifies 3D objects in a stack [8], similarly to the 2D analysis performed by Analyze> Analyze Particles… It is complemented by 3D Roi Manager [51], a companion plugin that adds a 3D ROI Manager to ImageJ.

3D Viewer

3D Viewer brings powerful hardware-accelerated 3D visualization to ImageJ [101], extending the limited functionality of Image>Stacks> 3D Project… In the ImageJ 3D Viewer stacks can be displayed as texture-based volume renderings, surfaces or orthogonal. It is macro-recordable and can be used by other plugins as a high-level programming library for 3D visualization.

Simple Neurite Tracer

Simple Neurite Tracer allows semi-automated segmentation of tubular structures in 3D [71].

TrakEM2

As mentioned earlier, TrakEM2 features powerful tools for multi-dimensional regions of interest [13].

See also:

Image>Stacks>3D Project…/Orthogonal Views [H], Analyze>Surface Plot…, XXI Skeletonize vs Skeletonize 3D, 3D tools in Fiji, Three Pane Crop, 3D image processing tutorials on the ImageJ wikibase.
Part III
Extending ImageJ

ImageJ capabilities can be extended by loadable code modules in the form of macros, scripts or plugins. 300+ macros, 500+ plugins and 20+ scripts are available through the ImageJ web site. Below is a short description of these three type of ImageJ add-ons:

Macros  The easiest way to execute a series of ImageJ commands. The ImageJ macro language – a Java-like language – contains a set of control structures, operators and built-in functions and can be used to call built-in commands and other macros. Macro code is stored in text files (*.txt and *.ijm extensions).

Plugins  Much more powerful, flexible and faster than macros (most of ImageJ’s built-in menu commands are actually plugins) but harder to write and debug. Plugins are written in the Java programming language (*.java source files) and compiled to *.class files.

Scripts  ImageJ uses the Mozilla Rhino interpreter to run JavaScripts. Similarly to plugins, scripts have full access to all ImageJ and Java APIs but do not need to be compiled (scripts and macros run interpretively). On the other hand, scripts lack the simplicity of macro language and feel less integrated in ImageJ.

14 Macros

A macro is a simple program that automates a series of ImageJ commands. The easiest way to create a macro is to record a sequence of commands using the command recorder (Plugins> Macros> Record...).

A macro is saved as a text file (*.txt or *.ijm extension) and once installed executed by selecting the macro name in the Plugins>Macros submenu, by pressing a key or, in the case of Macro tools, by clicking on an icon in the ImageJ toolbar. In addition, any macro file placed in $ImageJ/plug-ins with an *.ijm extension will be installed in the Plugins menu like any other plugin (before version 1.41 only files with an underscore in the name would be listed).

There are more than 300 example macros, on the ImageJ Web site. To try one, open it in a browser window and drag it directly to the Main ImageJ window or, copy it to the clipboard (Ctrl A, Ctrl C), switch to IJ, and run File>New System Clipboard (Ctrl V, Ctrl Shift V), pasting the macro into a new Editor window. Run it using the editor’s Macros>Run Macro command (Ctrl R). Most of the example macros are also available in the macros folder, inside the ImageJ folder.

Macro Programming

The ImageJ community has created excellent tutorials on macro programming. These resources are indispensable guides to the ImageJ macro language:


2. The Built-in Macro Functions webpage (Help>Macro Functions... and Macros>Function Improved in IJ) is the indispensable guide to the built-in functions that can be called from the ImageJ macro language. It is thoroughly documented and constantly updated: http://imagej.nih.gov/ij/developer macro/functions.html


See also: Scripts, Plugins, Editor, Fiji Script Editor

15 Scripts

JavaScript scripting was introduced in ImageJ 1.41 in order to bring full access to ImageJ and Java APIs (see Advantages and disadvantages of JavaScript). ImageJ uses the Mozilla Rhino interpreter built into Java 1.6 for Linux and Windows to run JavaScript. Mac users, and users of earlier versions of Java, must download JavaScript.jar into the plugins folder. This JAR file is available on the ImageJ website and is included with the Mac version of ImageJ in $ImageJ/plugins/jars.

Example JavaScript programs are available at imagej.nih.gov/ij/macros/js/. Thread safe JavaScript code can be generated using the Recorder (Plugins>Macros>Record...). Scripts can be opened in the editor as any other macro. Scripts with the extension *.js can be run using Macros>Run Macro otherwise Macros>Evaluate JavaScript (Ctrl J) must be used.

JavaScript Programming

Resources on ImageJ JavaScript scripting include:


See also: Macros, Plugins, Editor, Fiji Script Editor

16 Plugins

Plugins are a much more powerful concept than Macros and Scripts and most of ImageJ’s built-in menu commands are in fact implemented as plugins. Quoting Werner Bailey [3]:

See also: Plugins, Editor, Fiji Script Editor
### Advantages and disadvantages of JavaScript in ImageJ

A thorough comparison between different scripting languages is available on the Fiji webpage.

<table>
<thead>
<tr>
<th>JavaScript Advantages</th>
<th>JavaScript Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full access to ImageJ and Java APIs</td>
<td>Slower, especially starting up</td>
</tr>
<tr>
<td>Standardized</td>
<td>No equivalent of macro sets</td>
</tr>
<tr>
<td>Richer language (objects, ? operator, break, continue, etc.)</td>
<td>Cannot use most of ImageJ’s 360+ built in macro functions</td>
</tr>
<tr>
<td>Extensive documentation</td>
<td>Requires knowledge of complex ImageJ and Java APIs</td>
</tr>
</tbody>
</table>

1. CodeBar is a convenient ‘ActionBar’ that retrieves snippets and common tasks frequently used in macro writing. ‘ActionBars’ provide one or many easy to use button bar(s) that extend ImageJ’s graphical user interface. You can read more about the ActionBar plugin at the ImageJ Documentation Portal.

Plugins are implemented as Java classes, which means that you can use all features of the Java language, access the full ImageJ API and use all standard and third-party Java APIs in a plugin. This opens a wide range of possibilities of what can be done in a plugin.

The most common uses of plugins are filters performing some analysis or processing on an image or image stack and I/O plugins for reading/writing not natively supported formats from/to other devices. But as you can see when looking at the plugins listed on the ImageJ plugins page, there are many other things you can do with plugins, such as rendering graphics or creating extensions of the ImageJ graphical user interface.

Plugins in the `ImageJ/plugins/` folder are listed at the bottom of the Plugins menu (see VII Organizing Commands in the Menu Bar). Only `.class` and `.jar` files in the plugins folder with at least one underscore in their name will be installed. Note that, with IJ1.44d an later, ImageJ no longer automatically installs, at startup, plugins in JAR file directories that start with a lower case letter.

### Developing ImageJ Plugins

More information on how to develop ImageJ plugins can be obtained on the following documents:

1. Developer Resources Page on the ImageJ website (Help ⊿ Dev. Resources...):
2. Dedicated tutorials on Fiji’s webpage:
3. Dedicated tutorials on the ImageJ Documentation Portal:
   [http://imagejdocu.tudor.lu/](http://imagejdocu.tudor.lu/)
4. Dedicated tutorials on the ImageJDev webpage:
   [http://developer.imagej.net/ides](http://developer.imagej.net/ides)

**SEE ALSO**: Macros, Scripts, Editor, Fiji Script Editor

---

### Running ImageJ From the Command Line

17 Scripting in Other Languages

Support for other languages is possible in ImageJ using Fiji and its powerful editor. Fiji adds extra support for BeanShell, Clojure, Python and Ruby. The following documents will introduce you to the advanced scripting capabilities of Fiji:

1. The extensive tutorial on scripting Fiji with Jython by Albert Cardona:
2. Dedicated tutorials on the Fiji webpage:

### Fiji Script Editor

Fiji features a more powerful script editor than ImageJ’s built-in Editor. The Fiji editor is an invaluable help when writing scripts in any of Fiji’s supported languages, including the ImageJ macro language. The editor features full undo support, syntax highlighting, tabs, bookmarks and several other tools that simplify scripting workflows in ImageJ. For more information visit Fiji’s editor website at [http://fiji.sc/wiki/index.php/Script_Editor](http://fiji.sc/wiki/index.php/Script_Editor).

The Fiji Script Editor (ImageJ A 1.44m). The Fiji Editor is an advanced text editor, supporting BeanShell, Jython, JRuby and other scripting languages. It does not support Function Finder... [F] but selecting a built-in macro function and running `Tools ⊿ Open Help on Macro Functions...` retrieves the documentation for the selected function.

**SEE ALSO**: Scripting in Other Languages, Running ImageJ From the Command Line, IJ _ED_, a plugin by Jérôme Mutterer that binds jEdit to ImageJ

### 18 Running ImageJ from the Command Line

ImageJ was devised as a desktop application. It can, however, run without a graphics environment (headless mode) by adding a special library (`headless.jar`) to the `ij.jar` classpath that overrides...
Running ImageJ From the Command Line

key ImageJ classes to work better headlessly. As described on the Fiji website, this strategy is implemented in Fiji through the --headless command line flag (see also Running ImageJ in headless mode and Using Cluster for Image Processing with IJ). Headless operations are simplified in ImageJ2.

ImageJ recognizes the following command line options:

- "file-name" Opens a file. Examples: 
  - blobs.tif
  - /Users/wayne/images/blobs.tif
  - e81*.tif

- -ijpath path Specifies the path to the directory containing the plugins directory. Example:
  - -ijpath /Applications/ImageJ

- -port Specifies the port ImageJ uses to determine if another instance is running. Examples:
  - -port1 (use default port address + 1)
  - -port2 (use default port address + 2)
  - -port0 (do not check for another instance (OtherInstance)

- -macro path [arg] Runs a macro or script, passing it an optional argument, which can be retrieved using `getArgument()` . Examples:
  - -macro analyze.ijm
  - -macro /Users/wayne/images/stack1

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

- -eval "macro code" Evaluates macro code. Examples:
  - -eval "print('Hello, world');"
  - -eval "return getVersion();"

- -run command Runs an ImageJ menu command. Example:
  - -run "About ImageJ..."

- -debug Runs ImageJ in debug mode.

See also: Linux installation, ImageJ Documentation Portal: Command line

Part IV

ImageJ User Interface

Unlike most image processing programs ImageJ does not have a main work area. ImageJ's main window is actually quite parsimonious containing only a menu bar (at the top of the screen on the Mac) containing all the Menu Commands, a Toolbar, a Status bar and a Progress bar. Images, histograms, profiles, widgets, etc. are displayed in additional windows. Measurement results are displayed in the Results Table. Most windows can be dragged around the screen and resized.

The ImageJ window (version 1.46j).

<table>
<thead>
<tr>
<th>Status bar</th>
<th>Progress bar</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Rectangular Selection Tool and Rounded</td>
<td>8 Wand Tool</td>
</tr>
<tr>
<td>2 Rectangular Selection Tool</td>
<td>9 Text Tool</td>
</tr>
<tr>
<td>3 Oval Selection Tool, Elliptical Selection Tool and Brush Selection Tool</td>
<td>10 Magnifying Glass</td>
</tr>
<tr>
<td>4 Polygon Selection Tool</td>
<td>11 Scrolling Tool</td>
</tr>
<tr>
<td>5 Freehand Selection Tool</td>
<td>12 Color Picker Tool</td>
</tr>
<tr>
<td>6 Straight Line Selection Tool, Segmented Line Selection Tool, Freehand Line Selection Tool and Arrow Tool</td>
<td>13 More Tools Menu</td>
</tr>
<tr>
<td>7 Angle Tool</td>
<td>A–H Customized tools installed from StartupMacros.txt, macros/toolsets, macros/tools or plugins/Tools</td>
</tr>
<tr>
<td>8 Point Tool and Multi-point Tool</td>
<td></td>
</tr>
</tbody>
</table>

Toolbar

The ImageJ toolbar contains tools for making selections, drawings, zooming and scrolling, etc. In addition, the right-side of the toolbar contains seven slots that can host any of the 60+ tools and 15+ toolsets available on the ImageJ website (see Custom Tools).

All ImageJ tools share common features:

- The button on the bottom right corner of some icons in the toolbar depicts a contextual menu that can be accessed by right-clicking on the tool icon (e.g., Stacks Menu).
- If an 'Options' dialog is available for a particular tool, it can be accessed by double clicking on the tool icon (e.g., Wand Tool).

Status bar

When the cursor is over an image, pixel intensities and coordinates are displayed in the status bar. After running a filter, elapsed time and processing rate (in pixels/second) are also displayed. When clicking on the status bar the ImageJ version, the Java version, memory in use, memory...
available and percent memory used will be displayed. As Selections are created or resized, selection properties (e.g., location, width, etc.) are displayed on the status bar.

In addition, clicking on ImageJ’s status bar, forces the Java garbage collector to run, which may help to reclaim unused memory (see Edit > Options > Memory & Threads…). You can assess this by running Plugins > Utilities > Monitor Memory…; each click on the Status bar should lead to a spike in the ImageJ’s memory utilization.

See also: Plugins > Utilities > ImageJ Properties..., Help > About ImageJ...

V Toggling Calibrated Units
If a spatial scale has been defined in Image > Properties... [P] or Analyze > Set Scale..., selection properties are displayed in the Status bar in calibrated units. Resizing or moving while holding down Alt forces this information to be displayed in pixels.

Progress bar

The progress bar, located to the right of the status bar, shows the progress of time-consuming operations. It will not appear if the operation requires less than approximately one second.

19 Tools

19.1 Area Selection Tools

These tools share the first four toolbar slots. As described in Toolbar, use the right click drop-down menu to switch a different tool. Selection Color can be changed by double clicking on the Point Tool/Multi-point Tool.

19.1.1 Rectangular Selection Tool

Location, width, height, and aspect ratio are displayed in the status bar during drawing (see V Toggling Calibrated Units).

Modifier keys:
- Shift: Selection is constrained to a square
- Alt: Current aspect ratio is maintained while resizing
- With arrow keys, width and height are changed one pixel at a time
- Ctrl: Selection is resized around the center

See also: Rounded Rectangular Selection Tool, Specify..., XVII Embedding Color Annotations in Grayscale Images, Toolbar Shortcuts

19.1.2 Rounded Rectangular Selection Tool

This tool creates rectangular shapes with rounded corners. It shares the same toolbar slot and the same modifier keys with the Rectangular Selection Tool. Double clicking on its icon opens the depicted dialog in which it is possible to specify:

- Stroke width: The width of the contour.
- Corner diameter: The arc size at the vertices.

Stroke/Fill Color: The contour (stroke) color or the filling color of the rounded rectangle. As explained in Edit > Selection > Properties..., selections can be either filled or contoured, but not both. The nine default selection colors (black, blue, cyan, green, magenta, orange, red, white, yellow) can be typed as text. Any other color must be typed in hex notation (see XIX Hexadecimal Color Values).

See also: Rectangular Selection Tool, XVII Embedding Color Annotations in Grayscale Images, Toolbar Shortcuts

19.1.3 Oval Selection Tool

Location, width, height, and aspect ratio are displayed in the status bar during drawing (see V Toggling Calibrated Units).

Modifier keys:
- Shift: Selection becomes circular
- Alt: Current aspect ratio is maintained while resizing
- With arrow keys, width and height are changed one pixel at a time
- Ctrl: Selection is resized around the center

See also: Elliptical Selection Tool, Specify..., V Toggling Calibrated Units, XVII Embedding Color Annotations in Grayscale Images, Toolbar Shortcuts

19.1.4 Elliptical Selection Tool

Ellipse properties are adjusted by dragging the four handlers on its antipodal points [4]. To rotate or resize, drag the handlers on its major axis (transverse diameter). To adjust eccentricity, drag the handlers on its minor axis (conjugate diameter).

See also: Oval Selection Tool, XVII Embedding Color Annotations in Grayscale Images, Toolbar Shortcuts
19.1.5 Brush Selection Tool

Adjusts (refines) the shape of area selections using a circular ‘brush’ [5]. Clicking inside the area selection and dragging along its boundary will expand the boundary outwards. Clicking outside the area selection and dragging along its boundary will shrink the boundary inwards. Once the tool has been applied, ImageJ will treat the adjusted ROIs as Composite Selections. The brush diameter can be adjusted by double clicking on the tool icon.

Modifier keys:

- **Shift**: Holding Shift forces the Brush Selection Tool to add pixels to the selection
- **Alt**: Holding Alt forces the Brush Selection Tool to subtract pixels from the selection

**See also:** XIII Converting Composite Selections, Toolbar Shortcuts

19.1.6 Polygon Selection Tool

Creates irregularly shaped selections defined by a series of line segments. Segment length and angle are displayed in the status bar during drawing (see V Toggling Calibrated Units). To create a polygon selection, click repeatedly with the mouse to create line segments. When finished, click in the small box at the starting point (or double click), and ImageJ will automatically draw the last segment. The vertex points that define a polygon selection can be moved and modifier keys can be used to delete or add new vertexes to the polygon.

Modifier keys:

- **Shift**: Shift-clicking on an existing vertex of the polygon adds a new corner point, smoothing the polygon edge
- **Alt**: Alt-clicking on an existing vertex of the polygon removes it

**See also:** Segmented Line Selection Tool, Enlarge…, V Toggling Calibrated Units, XVII Embedding Color Annotations in Grayscale Images, Toolbar Shortcuts

19.1.7 Freehand Selection Tool

As with the polygon selection tool, ImageJ automatically draws the last segment. Location and intensity of starting pixel are displayed in the status bar during drawing.

**See also:** Freehand Line Selection Tool, Polygon Selection Tool, Enlarge…, V Toggling Calibrated Units, XVII Embedding Color Annotations in Grayscale Images, Toolbar Shortcuts

19.2 Line Selection Tools

Use these tools to create line selections. The three line selection tools share the same toolbar slot. As described in Toolbar, use the right click drop-down menu to switch between line tools.

Double click on any line tool to specify the line width by opening the Image>Adjust>Line Width… widget, on which is also possible to apply a cubic spline fit to a polyline selection. Check the Sub-pixel resolution checkbox in Edit>Options>Profile Plot Options… to create line selections with floating-point coordinates (see Selections With Sub-pixel Coordinates).

19.2.1 Straight Line Selection Tool

Length and line angle are displayed in the status bar during drawing (see V Toggling Calibrated Units).

Modifier keys:

- **Shift**: Forces the line to be either horizontal or vertical
- **Alt**: Keeps the line length fixed while moving either end of the line
- **Ctrl**: Forces the two points that define the line to have integer coordinates when creating a line on a zoomed image
- **Shift**: While moving either end of the line, the line is rotated/resized about its center

**See also:** Calibration Bar…, iSet Scale…, V Toggling Calibrated Units, XVII Embedding Color Annotations in Grayscale Images, Toolbar Shortcuts

19.2.2 Segmented Line Selection Tool

Works exactly as described for the Polygon Selection Tool: Create a segmented line selection by repeatedly clicking with the mouse. Each click will define a new line segment. Double click when finished, or click in the small box at the starting point. The points that define a segmented line selection can be moved or deleted, and new points can be added. Length and line angle are displayed in the status bar during drawing (see Toggling Calibrated Units).

Modifier keys:

- **Shift**: Shift-clicking on an existing vertex adds a new one, adding a new segment to the segmented line
- **Alt**: Alt-clicking on an existing vertex of the segmented line removes it

**See also:** Polygon Selection Tool, Freehand Selection Tool, V Toggling Calibrated Units, XVII Embedding Color Annotations in Grayscale Images, Toolbar Shortcuts

19.2.3 Freehand Line Selection Tool

Select this tool and drag with the mouse to create a freehand line selection.

**See also:** Freehand Selection Tool, Overlay Brush, V Toggling Calibrated Units, XVII Embedding Color Annotations in Grayscale Images, Toolbar Shortcuts

19.3 Arrow Tool

This tool shares the same toolbar slot with the Line Selection Tools and can also be installed on a dedicated toolbar slot using the More Tools Menu menu (see Arrow). Double clicking on the tool icon opens its Options prompt [6].
Being an annotation tool, arrows are created using foreground color (see Color Picker... [K]) and not selection color (see Point Tool). Width and Size (in pixels) can be adjusted by dragging the respective sliders or by direct input. Apart from the arrow styles listed here, a Headless option is also available. As for painting tools (Brush, Flood Fill and Pencil), the Color dropdown menu provides a convenient way to reset the foreground color to one of the default options.

As with any other selection, use arrows to non-destructively overlay by pressing [B] (Image>Overlay>Add Selection... [b]) or [D] (Edit>Draw [d]) to permanently draw the arrow on the image (see XVII Embedding Color Annotations in Grayscale Images when working with non-RGB images).

The same modifier keys described to the Straight Line Selection Tool apply to the arrow tool:

- **Shift**: Forces the line to be either horizontal or vertical
- **Alt**: Keeps the line length fixed while moving either end of the line
- **Ctrl**: While moving either end of the line, the line is rotated/resized about its center

**SEE ALSO**: Color Picker window, XVII Embedding Color Annotations in Grayscale Images, Brush, Overlay Brush, Pencil, Text Tool, Toolbar Shortcuts

### 19.4 Angle Tool

This tool allows you to measure an angle defined by three points. Double click on the angle tool icon to enable the measurement of reflex angles. The angle is displayed in the status bar while the selection is being created or adjusted. Press [M] (Analyze>Measure... [m]) to record the angle in the Results Table.

**SEE ALSO**: Toolbar Shortcuts

### 19.5 Point Tool

Use this tool to create a point selection, to count objects or to record pixel coordinates.

**Modifier keys**:

- **Shift**: Shift-clicking adds more points, creating a multi-point selection (see Multi-point Tool). Point count is displayed on the Status bar
- **Alt**: Alt-clicking on a point deletes it. Alt-clicking and dragging with the Rectangular Selection Tool or Oval Selection Tool deletes multiple points

**Double clicking on the point tool icon (or running Edit>Options>Point Tool...) displays its configuration dialog box.**

**Mark Width** If greater than zero, a mark of the specified diameter will be permanently drawn in the current foreground color (cf. Color Picker... [K]). Note that marks modify the image (it may be wise to work with a copy) and color marks are only available with RGB images (see XVII Embedding Color Annotations in GrayscaleImages).

**Auto-Measure** If checked, clicking on the image records the pixel location and intensity. Note that if Mark Width is not zero, every time a point selection is measured a mark will be painted (cf. Measure... [m]). If unchecked, Edit>Draw [d] can be used to paint the mark (Mark Width diameter) at the location of each point.

**Auto-Next Slice** If checked, ImageJ will automatically advance to the next stack slice. Note that this feature will only allow one point per slice.

**Add to ROI Manager** If checked, points will be automatically added to the ROI Manager...

**Label Points** If checked, each point selection will be displayed with an accompanying numeric label.

**Selection Color** Specifies Selection color, chosen from one of the nine default colors: red, green, blue, magenta, cyan, yellow, orange, black and white. The chosen color is highlighted in the center of the Point/MultiPoint Tool. It can also be specified using Edit>Options>Colors...

**SEE ALSO**: Multi-point Tool, Using a Keyboard Shortcut to Change Selection Color, Cell Counter plugin, Toolbar Shortcuts

### 19.6 Multi-point Tool

The Multi-point Tool selects multiple points behaving as the Point Tool when **Shift** is pressed. **Label Points** is checked and Auto-Measure and Auto-Next Slice are deselected. As described for the Point Tool, **Alt** can also be used to remove points. Similarly, when using Edit>Draw [d] marks are painted with the diameter of Mark Width.

**SEE ALSO**: Point Tool, Cell Counter plugin, Toolbar Shortcuts

### 19.7 Wand Tool

Creates a selection by tracing objects of uniform color or thresholded objects. To trace an object, either click inside near the right edge, or outside to the left of the object. To automatically outline and measure objects have a look, e.g., at the WandAutoMeasureTool macro.

To visualize what happens, imagine a turtle that starts moving to the right from where you click looking for an edge. Once it finds the edge, it follows it until it returns to the starting point. Note that the wand tool may not reliably trace some objects, especially one pixel wide lines, looking for an edge. Once it finds the edge, it follows it until it returns to the starting point. Note that this feature will only allow one point per slice.

**Double clicking on the wand tool icon (or running Edit>Options>Wand Tool...) opens the configuration dialog box in which three modes (4-connected, 8-connected or ‘Legacy’) plus a tolerance value can be set [T].**

---

33 Last updated: 2012/10/02

34 Last updated: 2012/10/02
The Wand Tool. 4/8-connected particles can be traced within an intensity range.

Tolerance The wand takes the pixel value where you click as an initial value. It then selects a contiguous area under the condition that all pixel values in that area must be in the range initial value – tolerance to initial value + tolerance.

4-connected Only the four neighbors of a pixel are considered neighbors. E.g., the wand does not follow a one-pixel wide diagonal line because the pixels of that line are not four-connected.

8-connected Each pixel is considered to have eight neighbors. So the wand follows a diagonal line if you click onto it. On the other hand, if you have an area of constant value dissected by a one-pixel wide diagonal line, the 8-connected wand will ‘jump over the line’ and include the other part of that area.

Legacy In this mode no neighbor is checked and no tolerance is used. This is the default mode of the Wand Tool in ImageJ 1.42 and earlier.

Modifier keys:

Shift Clicking and dragging while holding down the Shift key runs Image > Zoom > To Selection

Alt Image zooms out (right-click behavior)

SEE ALSO: XVIII Working with Zoomed Canvases, Zoom commands, Toolbar Shortcuts

19.8 Text Tool

Use this tool to add text to images. It creates text ROIs, rectangular selections containing one or more lines of text. Note the following when using the Text Tool:

- Font style and text alignment is specified in the Fonts widget, activated by double clicking on 
  or by running Edit > Options > Fonts. Text is drawn in foreground color (see Color Picker...)

- Use Alt to type special unit symbols such as μ (Alt M) or Å (Alt Shift A). Note that
  menu shortcuts require holding down Ctrl while using the Text Tool (see Using Keyboard Shortcuts)

SEE ALSO: XVII Embedding Color Annotations in Grayscale Images, Toolbar Shortcuts, Temporary Activation of a Tool

19.9 Magnifying Glass

Magnifies and reduces the view of the active image. Activate the tool and click on the image to zoom in. Right-click (or Alt-click) to zoom out. The current magnification is shown in the image’s title bar. Double click on the magnifying glass icon to revert to the image’s original magnification. As explained in Image > Zoom > In [+], there are 21 possible magnification levels: 3.1, 4.2, 6.3, 8.3, 12.5, 16.7, 25, 33.3, 50, 75, 100, 150, 200, 300, 400, 600, 800, 1200, 1600, 2400 and 3200%.

Modifier keys:

Shift Clicking and dragging while holding down the Shift key runs Image > Zoom > To Selection

Alt Image zooms out (right-click behavior)

SEE ALSO: XVIII Working with Zoomed Canvases, Zoom commands, Toolbar Shortcuts

19.10 Scrolling Tool

Allows you to scroll through an image that is larger than its window. You can temporarily activate this tool (except when using the Text Tool) by holding down the space bar.

SEE ALSO: XVIII Working with Zoomed Canvases, Toolbar Shortcuts

19.11 Color Picker

Sets the foreground drawing color by ‘picking up’ colors from any open image. Colors can also be picked up from the Color Picker (CP) window (Image > Colors > Color Picker... [K]) using any tool. In the icon, the ‘eye dropper’ is drawn in the current foreground color while the frame around it is drawn in the current background color. Edit > Draw [ ] and Edit > Fill [ ] use the foreground color. Edit > Clear, Clear Outside and Cut [ ] use the background color. Double clicking on the tool icon will display the Color Picker window.

Modifier key:

Alt Alt-clicking with the Color Picker Tool on the image canvas ‘picks-up’ background color

SEE ALSO: XVII Embedding Color Annotations in Grayscale Images, Toolbar Shortcuts, Temporary Activation of a Tool
19.12 **More Tools Menu**

The eight Toolbar slots between the Color Picker Tool and the Holding Alt makes the brush paint in background color.

**More Tools Menu** (IJ 1.46n). The menu lists tools from Startup Macros in ImageJ/macros/, Toolsets installed in ImageJ/macros/toolsets/, built-in tools loaded from ij.jar (Arrow, Brush, Developer Menu, Flood Filler, LUT Menu, Overlay Brush, Pencil, Spray Can and Stacks Menu) and Single Tools installed in ImageJ/plugins/Tools/. While toolsets replace all the eight slots in the toolbar, single tools are installed in the first available slot, or in the last slot if no free slots are available.

19.13 ☰ **Arrow**

Installs a copy of the Arrow Tool on the first available toolbar slot (or the last if no free slots are available), so that it can be accessed without the need of selecting it on the Line Selection Tools dropdown menu. Refer to the original Arrow Tool for details and modifier keys.

**SEE ALSO:** Color Picker window, XVII Embedding Color Annotations in Grayscale Images, Brush, Overlay Brush, Pencil, Text Tool, Toolbar Shortcuts

19.14 ☰ **Brush**

A freehand paintbrush tool that draws invasively (as opposed to the Overlay Brush that draws on a non-destructive image overlay (see Overlays and Image>Overlay> commands). Double clicking on the tool icon opens its Options dialog box in which is possible to specify the Brush width (in pixels) and Color.

Being an annotation tool, the paintbrush paints in foreground color as reflected its icon (see XVII Embedding Color Annotations in Grayscale Images when working with non-RGB images). The Color dropdown menu provides a convenient way to reset the foreground color to one of the default options, bypassing the need of opening the Color Picker window, evoked using Ctrl[K]. As previously described (see Undo and Redo), undo is restricted to last drawing step. The Brush and Pencil tools are in all similar, differing only on brush (stroke) size.

**Modifier keys:**

- **Shift** Shift-dragging on the canvas will adjust the brush size
- **Alt** Holding Alt makes the brush paint in background color

**SEE ALSO:** Overlay Brush, Pencil, Freehand Line Selection Tool, Color Picker window, XVII Embedding Color Annotations in Grayscale Images, Toolbar Shortcuts

19.15 ☰ **Developer Menu**

A drop-down menu collecting several online resources and commands that are useful when writing Macros, Plugins or troubleshooting ImageJ operations.

**Debug mode** activates ImageJ’s debugging mode (Edit>Options>Misc...).

**SEE ALSO:** Extending ImageJ, Editor, Help, Plugins, Macros, Utilities, New, Common Commands Menu Tool, Stacks Menu, LUT Menu

---

**SEE ALSO:** Custom Tools, Toolbar Shortcuts
19.16 Flood Filler

A paint bucket tool that fills with the current foreground color adjacent pixels that have the same value as the clicked pixel. Double-click on the tool icon to specify the flood type in terms of pixel connectivity: 4-connected or 8-connected.

To spread the fill to contiguous pixels within an intensity range, use the Wand Tool instead: Double-click on the Wand Tool icon to set a Tolerance value, then press (F) (Edit ➤ Fill) to fill with foreground color (highlighted in the Flood Filler icon) or (Backspace Del) (Edit ➤ Clear) to fill with background color (see Color Picker... [K]).

Modifier keys:

Alt Alt-clicking makes the brush paint in background color

See also: floodFill(x,y) macro function, Color Picker window, XVII Embedding Color Annotations in Grayscale Images, Toolbar Shortcuts

19.17 LUT Menu

A drop-down menu listing all the Image ➤ Lookup Tables commands. It is a convenient way to deal with a large collection of lookup tables that otherwise would only be accessed through the menu bar. Note that although it is not possible to organize LUTs into subfolders, it is possible to rename the most frequently used lookup tables with a numeric prefix (e.g. 01-glasbey.lut, 02-Termal.lut, etc.) so that they are listed earlier in the menu.

See also: Pseudocolor Images, Show_All_LUTs (a macro that creates a graphical palette of all the installed lookup tables), Stacks Menu, Common Commands Menu Tool, Developer Menu

19.18 Overlay Brush

A freehand paintbrush that draws on a non-destructive image overlay (see Overlays), as opposed to the Brush tool that draws invasively over the canvas.

Double-clicking on the tool icon opens its Options dialog box in which is possible to specify the Brush width (in pixels), Transparency (%) and Color.

As previously described (see Brush and Pencil tools), the Color dropdown menu changes the foreground color, bypassing the Color Picker window (activated by [C] [K]). Press Undo to remove the last painted stroke from the overlay. Overlay manipulations are described in Image ➤ Overlay ▶.

See also: Frehand Line Selection Tool, XVII Embedding Color Annotations in Grayscale Images, Toolbar Shortcuts

19.19 Pencil

A freehand painting tool that draws invasively in foreground color. It is in all similar to the Brush tool but it is typically used with thinner strokes. Double-clicking on the tool icon opens its Options dialog box in which is possible to specify the Pencil width (in pixels) and Color. Refer to the Brush tool tools for details.

Modifier keys:

Shift Shift-dragging on the canvas will adjust the brush size
Alt Holding Alt makes the brush paint in background color (see Color Picker... [K])

See also: Frehand Line Selection Tool, Overlay Brush, Color Picker window, XVII Embedding Color Annotations in Grayscale Images, Toolbar Shortcuts

19.20 Pixel Inspector

The Pixel inspector displays the values of a square neighborhood around the current cursor position as a table [9]. Values are updated in real time as the mouse is dragged over the image. It is useful to examine how a filter changes the pixel data. E.g., load Pixel Inspector, move the cursor over an image and run Process ➤ Filters ➤ Gaussian Blur...: When toggling the Preview checkbox you will be able to monitor in real time how different Sigma radius change pixel values.

In the Pixel Values table, columns and row headers (x & y positions) are expressed in pixel coordinates. The y-axis direction is determined by the Invert Y coordinates value in Analyze ➤ Set Measurements... The center position (current cursor) is printed in red (x, y, value). When the table is in the foreground, the arrow keys can be used to nudge the neighborhood square (outlined in red) and the table can be copied into the clipboard by pressing C. For settings, press the Pref button at the top left of the table:

Radius Specifies the size of the table, 3 × 3 for radius = 1; 5 × 5 for radius = 2, etc.
Grayscale readout The numeric output for grayscale images. Can be Raw [the default], Calibrated [see Analyze ➤ Calibrate...] or Hexadecimal (Hex). The mean grayscale value is determined by the weighting factors specified in Edit ➤ Options ➤ Conversions.
RGB readout The numeric output for RGB images. Can be R,G,B triplets, Gray Value or Hexadecimal (Hex) [see XIX Hexadecimal Color Values]. The mean grayscale value is determined by the weighting factors specified in Edit ➤ Options ➤ Conversions.
Copy to clipboard Specifies which data is copied to the clipboard. Choose Data only to copy the table without headers, x, y and Data to copy the current position (x, y) values followed by remaining data or Header and Data to copy the table with headers. Tables are copied as tab-delimited values.

See also: Text Images, Image ➤ Transform ➤ Image to Results/Results to Image, File ➤ Save As ➤ Text Image..., Import ➤ Text Image..., Toolbar Shortcuts
Spray Can

The Spray Can (Airbrush tool) draws random pixels in the current foreground color (paint) (see Color Picker... [6] and XVII Embedding Color Annotations in Grayscale Images). It behaves as a traditional airbrush or spray paint: Holding the main mouse button (without moving the cursor) will build up paint, as if pressing the nozzle of an aerosol paint can. Spray width, Dot size and Flow rate can be specified by double clicking on the tool icon.

This tool is useful to generate random spot noise. Use it to, e.g., assess the effectiveness of median filtering. Load the Spray Can tool, apply it over an image and toggle the Preview option in the Process>Filters>Median... prompt.

See also: Process>Noise>Add Noise, Salt and Pepper, Toolbar Shortcuts

Stacks Menu

A drop-down menu collecting several commands related to Stacks and Hyperstacks, otherwise accessed through the hierarchy of Image>Stacks>, Image>Hyperstacks> and File>Open Samples>. The list makes a particular emphasis on commands that have no keyboard shortcuts assigned.

See also: Plugins>Shortcuts>, LUT Menu, Stacks Menu, Common Commands Menu Tool, Developer Menu

Custom Tools

Customized tools are add-ons (macros and plugins) that allow custom interactions with the ImageJ toolbar and/or the image canvas. They are installed on the right side of the Toolbars between the Color Picker Tool and the More Tools Menu. At startup, the default set of tools is loaded from ImageJ/macros/StartupMacros.txt. Later on, tools can be appended or replaced using the More Tools Menu menu. As mentioned, custom tool configurations are saved in the preferences file, and thus remembered across restarts (see Settings and Preferences).

It is worth it to mention some differences between the installation of single tools and toolsets:

Single Tools Single tools are appended to the first available toolbar slot or installed in the last slot if no free slots are available. Tools can be macros (i.e., .txt and .ijm files) or plugins (i.e., .class and .jar files) and are listed on the More Tools Menu menu if placed in the ImageJ/macros/Tools/ directory. In addition to the macro tools distributed with ImageJ and saved in ImageJ/macros/tools/, a vast repertoire of tools is available on the ImageJ website.

Toolsets Toolsets are macro files (.txt and .ijm files) containing up to eight macro tools, along with any number of ordinary macros. Toolsets are listed on the More Tools Menu menu if installed in the ImageJ/macros/toolsets/ directory. Choosing a toolset (e.g., Lookup Tables) replaces all previously installed tools.

As mentioned, ImageJ/macros/StartupMacros.txt contains the tools loaded at startup. This file can be customized using Plugins>Macros>Startup Macros... or by holding Shift when choosing Startup Macros from the More Tools Menu menu.

ImageJ feature several pre-installed toolsets [8] and many others are available on the ImageJ website. Toolsets can also be created by choosing Toolset Creator, a convenient way to create groups of Menu Tools listing Plugins> commands.

21 Contextual Menu

As mentioned earlier macros and macro tools in the StartupMacros.txt are automatically installed in the Plugins>Macros> submenu and in the toolbar when ImageJ starts up.

In addition, the StartupMacros.txt file also installs the contextual (popup) menu displayed when right-clicking on an image. Other macros and toolsets (e.g., Magic Montage) may also replace the default menu with specialized ones. In this case, re-installing the StartupMacros (using the More Tools Menu) will revert the contextual menu to its default.

The ImageJ Macro Language — Programmer’s Reference Guide explains how this menu can be customized:

The menu that is displayed when a user right-clicks (or ctrl-clicks) on an image window can be customized through installation of the "Popup Menu" macro. Any menu has a name and a list of menu items. The newMenu(name, items) macro function allows the creation of a new menu. This menu passes the chosen item as a simple string to the "Popup Menu" macro. From this point you can decide what to do, according to what item was chosen.

Customizing the Image Popup Menu

/* The "Popup Menu" macro defines the menu that is displayed when right-clicking (or ctrl-clicking) on an image. It is part of the startup macros (StartupMacros.txt) and several other macro toolsets */


macro "Popup Menu" {
    var cmd = getArgument();
    if (cmd == "Help...") {
        showMessage("About Popup Menu");
    } else {
        run(cmd);
    }
}

So, e.g., to add the ability to run the Process>Subtract Background... command from the contextual menu one can simply add that command to the list of items defining the Popup Menu. Note that "-" defines menu separators.
Results Table

<table>
<thead>
<tr>
<th>Label</th>
<th>Area</th>
<th>Mean</th>
<th>StdDev</th>
<th>Mode</th>
<th>Skew</th>
<th>Kurtosis</th>
<th>Min</th>
<th>Max</th>
<th>XM</th>
<th>YM</th>
<th>XML</th>
<th>YML</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ImageJ Results table (version 1.44k). Columns width can be adjusted by clicking on and dragging the vertical lines that separate the column headings. Selected rows can be deleted by pressing the backspace key. The arrow keys can be used to vertically scroll the window.

22 Results Table

Most of ImageJ analyses are printed to the Results table. Table commands are organized in four menus: File ▹, Edit ▹, Font ▹ and Results ▹. A contextual menu listing the majority of these commands can be accessed by right-clicking anywhere in the Results window.

File ▹ Save As... Exports the measurements as a tab-delimited or comma-delimited text file as defined in Results ▹ Options... .

File ▹ Rename... Renames the table. Because ImageJ outputs measurements exclusively to the Results table, renaming the table will freeze its contents.

File ▹ Duplicate... Creates a new table containing a copy of the data. Note that ImageJ will not output measurements to duplicated tables.

Font ▹ This menu contains commands to adjust font size.

Results ▹ Clear Results... Alias for the Analyze ▹ Clear Results command.

Results ▹ Summarize Alias for the Analyze ▹ Summarize command.

Results ▹ Distribution... Alias for the Analyze ▹ Distribution... command.

Results ▹ Set Measurements... Alias for the Analyze ▹ Set Measurements... command.

Results ▹ Options... Opens the Edit ▹ Options ▹ Input/Output... dialog in which it is possible to specify if column headers and row numbers should be saved or copied from ImageJ tables (including the Summarize table, cf. Analyze ▹ Analyze Particles...). In addition, it allows to specify the file extension to be used when saving data. Custom extensions (e.g., .csv, .xls or .ods) allow ImageJ tables to be imported seamlessly by spreadsheet applications. ImageJ tables are saved in CSV format if File extension for tables is .csv.

See also: Plugins ▹ New ▹ Table...

The ImageJ editor (version 1.43n). The editor is a simple text editor featuring Function Finder... [F], an essential tool when writing Macros. The Fiji Script Editor is a more advanced editor featuring syntax highlight and support to all of Fiji's scripting languages.

23 Editor

Macros ▹, Scripts and Plugins can be opened and executed in the ImageJ editor. The editor commands are organized in five menus: File ▹, Edit ▹, Font ▹, Macros ▹ and Debug ▹.

File ▹ Basic file operations (Open, Save, Print, etc.) are listed in this menu. The last saving directory is kept in .fijiprefs.txt, the IJ preferences file (see Settings and Preferences).

Edit ▹ Similarly to any other text editor this menu contains commands related to text handling as well as commands for locating text. Specially useful are:

- Go to Line... [L]. This dialog box enables you to quickly go to a specified line of code.

- Zap Gremlins This command finds and deletes the extraneous non-visible, non-printing characters that sometimes appear when cutting and pasting from other sources, such as email messages that may contain extraneous control characters, or any non-ASCII characters.

- Copy to Image Info This command will copy the selected text (or the entire contents of the editor if no selection is present) to the image header, being available through the Image ▹ Show Info... [I] command. Note that the copied text will substitute any other information present in the file header and will only be available in images saved as TIFF (see II Image Types: Lossy Compression and Metadata).

Font ▹ This menu contains commands to adjust font size and type.

Macros ▹ This menu contains commands that allow you to run, install or evaluate macro code:

- Run Macro [R] [Ctrl R]. Runs the macro or the selected line(s) of code.

- Evaluate Line [Y] [Ctrl Y]. Runs the line of code that contains the insertion point.

- Abort Macro Exits the macro.
Install Macros [I] (Ctrl I). Adds the macro(s) contained in the editor to Plugins⇒Macros⇒subsubmenu (Plugins⇒Help⇒Macro Functions... command).

Macro Functions... [M] (Ctrl Shift M). Opens the Macro Functions reference page, the indispensable guide to the built-in functions that can be called from the ImageJ macro language (alias for Help⇒Macro Functions...).

Function Finder... [F] (Ctrl Shift F). Retrieves macro functions in the same way Find Commands... [I] retrieves commands. Functions are read from the functions.html file stored in the macros folder (a local copy of http://imagej.nih.gov/ij/developer/macro/functions.html). This file is deleted by Help⇒Update ImageJ... command every time ImageJ is updated to a release version (i.e., not a daily build, see Installing and Maintaining ImageJ), forcing Function Finder to download a fresh copy the next time it is launched.

Evaluate JavaScript [J] (Ctrl J). Runs JavaScript code in the editor window. Note that Run Macro runs JavaScript code if the title of the file ends with ‘.js’.

Debug⇒ This menu contains seven commands related to the macro debugging. You can debug a macro using the commands in the Debug menu. You start a debugging session initiating Debug Macro. You can then single step through the macro code by repeatedly running Step.

Debug Macro [D] (Ctrl D). Starts running the macro in debug mode and opens the ‘Debug’ window, which initially displays the memory usage, number of open images, and the active image’s title. The macro stops running at the first executable line of code, which is highlighted. Use one of the following commands to continue execution.

Step [E] (Ctrl E). Executes the highlighted statement and advances to the next. The variable names and values in the ‘Debug’ window are updated.

Trace [T] (Ctrl T). Runs the macro, displaying variable names and values in the ‘Debug’ window as they are encountered.

Fast Trace [T] (Ctrl Shift T). Same as above, but faster.

Run Runs the macro to completion at normal speed (similarly to Macros⇒Run Macro).

Run to Insertion Point (Ctrl Shift E). Runs the macro to a statement that was previously defined by clicking the mouse on an executable line of code.

Abort Exits debug mode.


24 Log Window

The Log window is used to display useful information about ongoing operations. It is frequent for plugins and macros to send messages to the Log window reporting progress, errors or troubleshooting information.

If you are troubleshooting a problem, you can check Debug mode in Edit⇒Options⇒Misc... to have ImageJ outputting messages to the Log window (ImageJ will exit debug mode as soon as the Log window is closed).

In addition, Tiff tags and information needed to import files are printed to the log window when ImageJ runs in Debug Mode.

Most of the general shortcut keys described in Editor apply to the Log window.

25 Customizing the ImageJ Interface

Most settings determining the look and feel of ImageJ are listed in Edit⇒Options..., namely Edit⇒Options⇒Appearance... and Edit⇒Options⇒Misc... (see also Settings and Preferences). However, other aspects of the ImageJ interface can also be personalized.

25.1 Floating Behavior of Main Window

It is possible to place the Main ImageJ window above all other windows at all time using a simple JavaScript instruction: IJ.getInstance().setAlwaysOnTop(true).

To test it, copy this one line script to the clipboard (or download Always_on_Top.js from the online scripts repertoire), switch to ImageJ, type Shift V (File⇒New⇒System Clipboard [V]), then type [Ctrl E] (Macros⇒Evaluate JavaScript [J]). To create an “Always on Top” command, save this script in the plugins folder as Always_on_Top.js and run Help⇒Refresh Menus to start using the new command. Macro (5) Customizing the Float Behavior of IJ’s Main Window exemplifies how to set this option at launch.

SEE ALSO: I, I Frontmost Window and Window Activation
Examples of modified crosshair pointers, more visible on grayscale images. The default crosshair cursor can be replaced by any image saved as crosshair-cursor.gif in ImageJ/images/.

25.2 Pointer

At startup, ImageJ will search for a GIF image named crosshair-cursor.gif in the ImageJ/images/ directory. If present, it will be used to replace the default crosshair cursor. The pointer can also be changed to an arrow by toggling Use pointer cursor on Edit > Options > Misc...

(5) Customizing the Float Behavior of IJ’s Main Window

// These macros can be added to the ImageJ/macros/StartupMacros.txt file in order to set the floating behavior of the ImageJ main window

// option 1) Run ImageJ/plugins/Always_on_Top.js command at launch, by adding it to the "AutoRun" macro
macro "AutoRun" {
  run("Always on Top");
}

// option 2) Execute the script at launch, by adding it to "AutoRun"
macro "AutoRun" {
  eval("script", "IJ.getInstance().setAlwaysOnTop(true)");
}

// option 3) Toggle the setAlwaysOnTop option using a shortcut, e.g., F1
var afloat;
macro "Toggle AlwaysOnTop [F1]" {
  boolean[] booleans = new boolean[] {true, false};
  eval("script", "IJ.getInstance().setAlwaysOnTop("+ booleans[afloat] +")");
  afloat = !afloat;
}

Part V
Menu Commands

As described in ImageJ User Interface, the menu bar lists all ImageJ commands. It is organized in eight menus:

- File - Basic file operations (opening, saving, creating new images). Most are self-explanatory.
- Edit - Editing and drawing operations as well as global settings.
- Image - Conversion and modification of images including geometric transformations.
- Process - Image processing, including point operations, filters and arithmetic operations.
- Analyze - Statistical measurements, profile and histogram plotting and other operations related to image analysis.
- Plugins - Commands for creating, editing and managing add-ons (see Extending ImageJ), listing all the user-installed Macros, Scripts and Plugins installed in the ImageJ/plugins/ directory.
- Window - Selection and management of open windows.
- Help - Updates, documentation resources and version information.

VII Organizing Commands in the Menu Bar

The Plugins menu can become easily cluttered after the installation of several plugins. Since Plugins reflects the hierarchy of directories in ImageJ/plugins/ (up to two subfolders), submenus (i.e., subfolders) can be created to keep the menu organized, preventing it from running off the bottom of the screen. E.g., to move the EPS Writer plugin into a Plugins > Input-Output > PDF > submenu, one would move EPS_Writer.class into ImageJ/plugins/Input-Output/PDF/.

In addition, checking the Move isolated plugins to Misc. menu checkbox in Edit > Options > Misc... will compact the menu list by moving to Plugins > Miscellaneous all the plugins with only one command that try to install themselves in submenus.

Note that external plugins can be installed in any of the ImageJ menus. This is the case of plugins packaged in JAR files containing a configuration file plugins.config specifying the location of the new commands implemented by the plugin. You can rename, reorganize or move commands implemented by external plugins by editing their plugins.config file as described on the JAR demo documentation page. If you don’t know in which menu a plugin has been registered, use Show full information in the command Finder (Plugins > Utilities > Find Commands... [l]) to find out the location of the installed .jar files.

With Fiji, Scripts and Macros can be registered in any menu by saving into \\Fiji.app\plugins\Scripts\menu name\submenu name/. E.g., a certain macro .ijm file saved in \\Fiji.app\plugins\Scripts\File\Import/ is registered in the File > Import > submenu.

See also: ImageJ’s plugin architecture on the Fiji website
26  **File**

26.1  **New**

Contains commands for creating new images, stacks, hyperstacks or text windows.

**SEE ALSO:**  Plugins › New

26.1.1  **Image... [n]**

Creates a new image window or stack. A dialog box allows you to specify the image title, type, dimensions and initial content.

*Name* is the title that will be used for the Window. *Type* is the image type: 8-bit grayscale, 16-bit grayscale (unsigned), 32-bit (float) grayscale or RGB color. *Fill With* (White, Black or Ramp) specifies how the image is initialized. *Width* and *Height* specify the image dimensions in pixels. Set *Slices* to a value greater than one to create a stack.

**SEE ALSO:**  Image › Hyperstacks › New Hyperstack... , Image Types and Formats

26.1.2  **Hyperstack...**

Alias for the Image › Hyperstacks › New Hyperstack... command.

26.1.3  **Text Window [N]**

Creates a new text window with the title ‘Untitled.txt’.

**SEE ALSO:**  Plugins › New › Text Window... , Macro , Table...

26.1.4  **Internal Clipboard**

Opens the contents of the internal ImageJ clipboard.

**SEE ALSO:**  Edit › Copy [c] , Cut [x] , Paste Control...

26.1.5  **System Clipboard [V]**

Opens the contents of the operating system clipboard.

**SEE ALSO:**  Edit › Copy to System , Cut [x] , Paste Control...

26.2  **Open... [o]**

Opens an image and displays it in a separate window. Image files must be in TIFF, GIF, JPEG, DICOM, BMP, PGM or FITS format, or in a format supported by a reader plugin. Also opens:

- ImageJ and NIH Image lookup tables (.lut extension).
- Tables (in tab-delimited text format) (.xls or .csv extension, see Results Table)
- Selections (.roi or .zip extension)
- Text files (.txt,.img,.js and .java extensions)

**SEE ALSO:**  File › Import... , Image Types and Formats, Virtual Stacks, VIII Opening Files: File › Open..., File › Import... and Drag & Drop

### VIII OPENING FILES: File › Open..., File › Import... and Drag & Drop

While the File › Open... [o] command opens formats natively supported by ImageJ (images and non-images files), the File › Import... submenu provides access to plugins for additional file types (e.g., reading ‘raw’ files, images in ASCII format or loading images over the network). Most of ImageJ’s Input/Output plugins are installed on this submenu.

Note that almost every format known to ImageJ can be opened by dragging and dropping the file into the Main ImageJ window. E.g., in the illustration below a remote macro file is opened by dragging its URL directly from a Web browser.

26.3  **Open Next [O]**

Closes the current image and opens the next image (if any) in its directory. Holding Alt opens the previous image (if any) in its directory.

26.4  **Open Samples**

Opens example images hosted on the ImageJ Web site. These sample images are useful for creating, testing and debugging macros since routines can be applied to the same image, regardless of where the macro is run. Among all, probably the most used is *blobs.gif*: Open Samples › Blobs (25K) [B].

Samples ➤ Cache Sample Images. The ‘AutoRun’ macro in the \texttt{StartupMacros.txt} file can then be used to change the default path of sample images, allowing a complete off-line usage of the \texttt{File ➤ Open Samples ➤} submenu:

```java
macro "AutoRun" {
    dir = getDirectory("imagej") + "samples";
    if (File.exists(dir)) {
        dir = replace(dir, " ", ";");
        if (startsWith(getInfo("os.name"), "Windows"))
            dir = "/" + replace(dir, File.separator, "/");
        call("ij.Prefs.setImageURL", "file://"+ dir +"/");
    }
}
```

26.5 Open Recent ➤

The submenu shows a list of the 15 recently opened files. Click on a filename to open it.

26.6 Import ➤

This submenu lists the installed image reader plugins.

See also: Non-native Formats, Acquisition plugins, Input/Output plugins, VirtualStackFromList macro, VIII Opening Files: File ➤ Open... ➤ File ➤ Import ➤ and Drag & Drop

26.6.1 Image Sequence... Opens a series of images in a chosen folder as a stack. Images may have different dimensions and can be of any format supported by ImageJ (see Image Types and Formats and HandleExtraFileTypes plugin). Non-image files (Scripts, \texttt{lut}, \texttt{.roi}, \texttt{RoiSet.zip}, etc.) are ignored.

Information – width \times \text{height} \times \text{depth} (size) – of the stack to be created is displayed at the bottom of the dialog.

\textbf{Number of Images} Specifies how many images to open.

\textbf{Starting Image} If set to \textit{n}, import will start with the \textit{n}th image in the folder.

\textbf{Increment} If set to ‘2’ every other image will be opened, if set to ‘3’ to every third image will be opened, etc.

\textbf{File Name Contains} Enter a string into this field and ImageJ will only open files whose name contains that string.

\textbf{Enter Pattern} Regular expressions (regex) can be typed here for advanced filtering (see Basic syntax of regular-expressions).

\textbf{Scale Images} Setting a value less than 100\% will reduce memory requirements. E.g., entering 50 reduces the amount of memory needed to open a stack by 25\% (two-dimensional images: \(0.5 \times 0.5 = 0.25\) of the original data). This value is ignored if Use Virtual Stack is checked.

\textbf{Convert to RGB} Allows a mixture of RGB and grayscale images to be opened by converting all the sequence to RGB. Note that if this option is unchecked and the first imported image is 8-bit then all the remaining images in the sequence will be converted to 8-bit. Checking this option, circumvents this issue.

\textbf{Sort Names Numerically} When checked, the stack will be opened in \textit{numeric} file name order (e.g., ‘name1.tif’, ‘name2.tif’, ‘name10.tif’) instead of alphanumeric order (e.g., ‘name1.tif’, ‘name10.tif’, ‘name2.tif’). DICOM files in the same series (tag\#0020,0011) are always sorted by the image number (tag\#0002,0013). The List Stack Tags macro, part of the ListDicomTags macro set, lists the values of the image number and image series tags.

\textbf{Use Virtual Stack} When checked, images are opened as a read-only virtual (disk-resident) stack using a version of the Virtual Stack Opener plugin. This allows image sequences too big to fit in RAM to be opened, but access time is slower and changes are lost when switching to a different image in the stack (see Virtual Stacks). Note the following consequences of enabling this option:

- Image Overlays are not loaded
- If the folder contains tiff stacks, only the first slice of those stacks will be imported (with RAM resident stacks, all slices are imported and concatenated into the sequence)

\textbf{Help} Opens \url{http://imagej.nih.gov/ij/docs/menus/file.html#seq1}.

See also: File ➤ Save As ➤ Image Sequence... ➤, OpenSeriesUsingFilter macro, Image ➤ Overlay... Regular-expressions basic syntax summary. For more information on regex filtering see \url{http://download.oracle.com/javase/tutorial/essential/regex/}.

<table>
<thead>
<tr>
<th>Regex Syntax (Character Classes)</th>
<th>Example</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ ] Delimit the set of characters to match</td>
<td>[aA]</td>
<td>Either lower or upper case A</td>
</tr>
<tr>
<td>- Character ranges</td>
<td>[0-9]</td>
<td>Any digit (from 0 through 9)</td>
</tr>
<tr>
<td>- Any character</td>
<td>[0-9]</td>
<td>A digit plus any other character</td>
</tr>
<tr>
<td>* Zero or more of the preceding item</td>
<td>*</td>
<td>Any character sequence</td>
</tr>
<tr>
<td>+ Zero or one of the preceding item</td>
<td>[0-9]?</td>
<td>An optional digit</td>
</tr>
<tr>
<td></td>
<td>[0-9]+</td>
<td>At least a digit</td>
</tr>
<tr>
<td>- Negation</td>
<td>[^0-9]</td>
<td>Any character that is not a digit</td>
</tr>
<tr>
<td>&amp;&amp; AND (Intersection)</td>
<td>[0-9]&amp;&amp;[^3]</td>
<td>A digit that is not 3</td>
</tr>
<tr>
<td></td>
<td>OR (Alternation)</td>
<td>[0-9]</td>
</tr>
</tbody>
</table>
IX Reducing Memory Requirements When Importing Images

Since ImageJ 1.44d, the File Import Image Sequence... command no longer features the Convert to 8-bit Grayscale checkbox. This option was used to reduce memory requirements but used different scaling for each imported image.

As a replacement, use the Use virtual stack option and then convert to 8-bit using File Type > 8-bit. Memory requirements can also be reduced by using the Scale Images (%) option. The amount of memory allocated to ImageJ can be adjusted in Edit Options > Memory & Threads...

26.6.2 Raw...

Use this command to import images that are not in a file format directly supported by ImageJ. You will need to know certain information about the layout of the image file, including the size of the image, and the offset to the beginning of the image data.

Interleaved RGB images have pixels stored contiguously (rgbrgb... ) in a single image plane. Planar RGB images have the red, green and blue image data stored in separate 8-bit sample planes. ImageJ saves RGB images (both TIFF and raw) in interleaved format.

Image Type There are fourteen choices depicted above. 16-bit signed integer images are converted to unsigned by adding 32,768. 1-bit Bitmap images are converted to 8-bit.

Image Width The number of pixels in each row of image data.

Image Height The number of rows in the image.

Offset to First Image The number of bytes in the file before the first byte of image data.

Number of Images The number of images stored in the file. If this value is greater than the actual number of images the resulting stack will get truncated to the actual size.

Gap Between Images The number of bytes from the end of one image to the beginning of the next. Set this value to width×height×bytes-per-pixel×n to skip n images for each image read.

White is Zero Should be checked if black pixels are represented using numbers that are less than the numbers used for white pixels. If your images look like photographic negatives, changing this field should fix the problem.

Little-Endian Byte Order Probably needs to be checked when importing 16-bit or 32-bit grayscale images from little-endian machines such as Intel based PCs.

Open All Files in Folder If checked, ImageJ will import all the images in the folder as a stack. The images must all be the same size and type.

Use Virtual Stack Images are imported as virtual stacks.


See also: Image Types and Formats

26.6.3 LUT...

Opens an ImageJ or NIH Image lookup table, or a raw lookup table. The raw LUT file must be 768 bytes long and contain 256 reds, 256 blues and 256 greens. If no image is open, a 256×32 ramp image is created to display the LUT. Note that lookup tables with file names ending in .lut can also be opened using File Open... [o] or drag and drop.

26.6.4 Text Image...

Opens a tab-delimited text file as a 32-bit real image (see Text Images). The image’s width and height are determined by scanning the file and counting the number of words and lines. For text files with integer values no larger than 255, use Image Type > 8-bit to convert to 8-bit. Before converting, disable Scale When Converting in Edit Options > Conversions... to prevent the image from being scaled to 0–255.

See also: Text Images, Pixel Inspector, Image Transform > Image to Results/Results to Image, Save As > Text Image..., Open TextImagesAsStack macro

26.6.5 Text File...

Opens a text file. Note that text files can also be opened using File Open... [o] or drag and drop.

26.6.6 URL...

Downloads and displays known formats to ImageJ specified by a URL. Other URLs ending with ‘/’ or ‘.html’ are opened in the user’s default browser. The Input URL is saved in the ImageJ preferences file and retrieved across IJ restarts.

It is also possible to open zip archives, using a URL that contain multiple DICOM images. Some example URLs are:

- file://Macintosh HD/images/Nanoprobes.tif
- file://D:images/neuron.tif

26.6.7 Results...

Opens an ImageJ table, or any tab or comma-delimited text file (see Results Table). Note that .csv and .xls files can also be opened by drag and drop.
26.6.8 Stack From List...

Opens a stack, or virtual stack, from a text file or URL containing a list of image file paths [11]. The images can be in different folders but they must all be the same size and type. The Virtual Stack From List macro demonstrates how to generate a list of images and then use that list to open the images as a virtual stack. The OpenStackUsingURLs macro demonstrates how to how to open an image series from a remote server.

26.6.9 TIFF Virtual Stack...

Opens a TIFF file as virtual stack (see Virtual Stacks and III Opening Virtual Stacks by Drag & Drop).

26.6.10 AVI...

Uses a built in version of the AVI reader plugin to open an AVI file (JPEG or PNG compressed, or uncompressed) as a stack or virtual stack (one slice per video frame) [12]. Animation speed is retrieved from image frame rate. AVI files can also be opened using File>Open... [o] or drag and drop but macros must use this command to gain access to the dialog box options. ImageJ supports a restricted number of AVI formats including MJPG (motion-JPEG) and various YUV 4:2:2/4:2:0 compressed formats (cf. plugin source code). The OME Bio-Formats library (see Non-native Formats) extends support to MSRLLE and MSV1 encoded formats.

The dialog prompt allows you to choose if frames should be converted to 8-bit grayscale or flipped vertically. For large files, an option to open the movie as a virtual stack is also available (see Virtual Stacks). It is also possible to specify the starting and ending frame. Enter 0 (zero) to specify the last frame, −1 to specify the second last frame, etc.

See also: File>Save As>AVI...

26.6.11 XY Coordinates...

Imports a two column text file, such as those created by File>Save As>XY Coordinates..., as a polygon selection. The selection is displayed in the current image or, if the current image is too small, in a new blank image. Coordinates of active selection (at evenly spaced one pixel intervals) can be retrieved using the List coordinates options in Edit>Selection>Properties... [y].

26.7 Close [w]

Closes the active image.

26.8 Close All

Closes all open images. An alert is displayed if there are unsaved changes.

26.9 Save [s]

Saves the active image in TIFF format, the ‘default’ format of ImageJ (cf. II Image Types: Lossy Compression and Metadata). To save only a selected area, create a rectangular selection and use the Image>Duplicate... [D] command. Note that Save [s] and File>Save As>Tiff... are redundant commands.

26.10 Save As>

Use this submenu to save the active image in TIFF, GIF, JPEG, or ‘raw’ format. Can also be used to save measurement results, lookup tables, selections, and selection XY coordinates.

26.10.1 Tiff...

Saves the active image or stack in TIFF format in redundancy with File>Save [s]. TIFF is the only format (other than ‘raw’) that supports all ImageJ data types (8-bit, 16-bit, 32-bit float and RGB) and the only format that saves spatial and density calibration. In addition Selections and Overlays are also saved in the TIFF header.

By default, 16-bit and 32-bit images are saved using big-endian byte order. Check Save TIFF and Raw in Intel Byte Order in the Edit>Options/Input/Output... dialog box to save using little-endian byte order.

See also: Native Formats, II Image Types: Lossy Compression and Metadata, X Warning on JPEG Compression

26.10.2 GIF...

Saves the active image in GIF format. RGB images must first be converted to 8-bit color using Image>Type>8-bit Color. The value to be used as the transparent index (0–255) can be set in the Edit>Options/Input/Output... dialog box. Stacks are saved as animated GIFs. Use Image>Stacks>Tools>Animation Options... [Alt/] (or right-click on the on the play/pause icon that precedes the stack slider) to set the frame rate.

26.10.3 Jpeg...

Saves the active image in JPEG format. Edit JPEG Quality Edit>Options/Input/Output... dialog box to specify the JPEG compression level (0–100). This value is shown on the title of the save dialog prompt. Lower values produce smaller files but poorer quality. Larger values produce larger files but better quality. Color sub-sampling is disabled when the value is set to 100, reducing the likelihood of color artifacts. By default, the DPI in the JPEG header is set to 72. For a higher value, use a unit of inch in the Analyze>Set Scale... dialog. E.g., setting Distance in Pixels to 300, Known Distance to 1 and Unit of Length to inch will set the DPI to 300.

Overlays are embedded when saving in JPEG format (see Image>Overlay>Flatten [F]).

See also: II Image Types: Lossy Compression and Metadata, X Warning on JPEG Compression
**X Warning on JPEG Compression**

The JPEG format uses Lossy compression that leads to severe artifacts that are not compatible with quantitative analyses. As such, it should only be used for presentation purposes (if file size is an issue), but even then a lossless format such as PNG is probably more suitable.

The illustration below exemplifies the consequences of saving images in a lossy format. To replicate it, open the Mandrill sample image (by drag and drop, or alternatively using File→Import→URL... and typing the image’s path, http://imagej.nih.gov/ij/images/baboon.jpg), duplicate it (Images→Duplicate... [D]), save the duplicate as JPEG (File→Save As→Jpeg...), run File→Revert [r] (so that the saved version is reloaded by ImageJ) and calculate the difference between the two images using Process→Image Calculator...

By adjusting the Brightness/Contrast... [C], you will notice that the imperceptible JPEG artifacts are most pronounced along regions of higher contrast changes. In addition to this edge artifact, the JPEG algorithm may shift colors to improve compression which may lead to artificial colocalization. These artifacts are intrinsic to the format and may persist even if JPEG Quality has been increased to 100% in Edit→Options→Input/Output... Once an image has been lossy compressed there is no way of reverting it to the original. Given all this, and since Metadata is poorly supported in lossy formats, it is unreasonable to use JPEG in image processing.

![Original, JPEG copy (75% quality), Difference]  

**26.10.6 Raw Data...**

Saves the active image or stack as raw pixel data without a header. 8-bit images are saved as unsigned bytes, unsigned 16-bit images are saved as unsigned shorts and signed 16-bit images (e.g., File→Open Samples→CT (420K, 16-bit DICOM)) are saved as signed shorts. 32-bit images are saved as floats and RGB images are saved in three bytes per pixel (24-bit interleaved) format. 16-bit and 32-bit (Float) images are saved using big-endian byte order unless Export Raw in Intel Byte Order is checked in the Edit→Options→Input/Output... dialog box.

**26.10.7 Image Sequence...**

Saves a Stack or a hyperstack as an image sequence.

**Format** Specifies the output format that can be set to either BMP, FITS, GIF, JPEG, PGM, PNG, Raw, Text Image, TIFF, or Zip (cf. Image Types and Formats). In IJ 1.44 and later, Overlays are embedded when saving in JPEG or PNG format.

**Name** Specifies the leading string that will be common to all numeric filenames.

**Start At** (Stacks only) Specifies the starting number of the sequence.

**Digits (1–8)** The number digits of the incremental sequence. Filenames are padded with leading zeros.

**Use slice labels as filenames** (Stacks only) If checked, each slice will be saved with its own label (the image subtitle displayed above the image, see Remove Slice Labels) and no numeric sequence will be used.

With hyperstacks, images are saved using ‘Name_ t d z d x d y d c d’ in which d is the incremental number of specified Digits; t, the frame; z, the slice and c, the channel, so e.g., for the depicted snapshot the first image would be saved as ‘mitosis_t001_z001_c001.tif’.

---

26.10.4 Text Image...

Saves the active image as a spreadsheet compatible tab-delimited text file. Calibrated and floating-point images are listed with the precision specified by Decimal places in Analyze→Set Measurements... For RGB images, each pixel is converted to grayscale using the formula $\text{gray} = (\text{red} + \text{green} + \text{blue})/3$ or the formula $\text{gray} = 0.299 \times \text{red} + 0.587 \times \text{green} + 0.114 \times \text{blue}$ if Weighted RGB to Grayscale Conversion is checked in Edit→Options→Conversions. See also: Text Images, Pixel Inspector, Image→Transform→Image to Results/Results to Image, Import→Text Image...

26.10.5 Zip...

Saves the active image or stack as a TIFF file inside a compressed ZIP archive.
26.10.8 AVI...

Exports a stack or hyperstack as an AVI file [12]. Note that AVI files larger than 2GB are not correctly written.

Compression JPEG, PNG or Uncompressed. With IJ 1.44 and later, Overlays are embedded when saving in JPEG or PNG format. The default compression is JPEG.

Frame Specifies the frame frequency. The proposed value is read from Image→Stacks→Tools→Animation Options… [Alt/] and Image→Properties… [P], as long as the unit of Frame Intersect is ‘sec’.

See also: File→Import→AVI...

26.10.9 PNG...

Saves the active image in PNG (Portable Network Graphics) format. All image types, except RGB, are saved as 8-bit PNGs. 16-bit images are saved as 16-bit PNGs. With 8-bit images, the value to be used as the transparent index (0–255, -1 for “none”) can be set in the Edit→Options→Input/Output… dialog box. Overlays are embedded when saving in PNG format.

26.10.10 FITS...

Saves the active image in FITS (Flexible Image Transport System) format [13].

26.10.11 LUT...

Saves the active image’s lookup table to a file. The 768-byte file consists of 256 red values, 256 green values and 256 blue values.

26.10.12 Results...

Exports the contents of the ‘Results’ window as a tab-delimited or comma-delimited (.csv) text file. Prior to ImageJ 1.44b this command used to be named ‘Measurements…’.

26.10.13 Selection...

Saves the current area selection boundary to a file, that can be later retrieved using File→Open… [o] to restore the selection. Active Selections and Overlays are saved in the TIFF header by default (see File→Save As→TIFF…).

26.10.14 XY Coordinates…

Exports the XY coordinates of the active ROI as a two column, tab-delimited text file. ROI coordinates can also be retrieved using the List coordinates option in Edit→Selection→Properties… [y], that tabulates ROI coordinates at evenly spaced one pixel intervals.

See also: Selections With Sub-pixel Coordinates, File→Import→XY Coordinates…

26.11 Revert [r]

Reloads the active image, stack or hyperstack from disk, reverting it to its last saved state. It is actually a shortcut for closing the window without saving, and then reopening it. Note that it may not work with Non-native Formats opened through external plugins such as the OME Bio-Formats library.

See also: Undo and Redo

26.12 Page Setup...

The Page Setup dialog allows you to control the size of printed output, plus other printing options:

- **Scale**: Values less than 100% reduce the size of printed images and values greater than 100% increase the size. 100% corresponds to 72 pixels per inch (ppi), about the unzoomed screen size of the image. The size of the printed image is determined by the Scale value and the width and height of the image in pixels. Spatial calibration is ignored.

  - **Draw border**: If checked, ImageJ will print a one pixel wide black border around the image.

  - **Center on page**: If checked, the image will be printed in the center of the page instead of in the upper left corner.

  - **Print title**: If checked, the title of the image will be printed at the top of the page.

  - **Selection only**: If checked, current selection will be printed instead of the entire image.

  - **Rotate 90°**: If checked, the image will be rotated 90° to the left before being printed.

  - **Print actual size**: Considers the DPI information in the image header (typically 72, cf. Jpeg…). For a higher value, use a unit of inch in the Analyze→Set Scale… dialog. E.g., setting Distance in Pixels to 300, Known Distance to 1 and Unit of Length to ‘inch’ will set the DPI to 300.

26.13 Print… [p]

Prints the active image. The size of the printed image will normally be slightly less its size on the screen (unzoomed). Use the Page Setup… dialog to increase of decrease the size of printed images. Images larger than the page are scaled to fit. Overlays are embedded when printing images.

26.14 Quit

Prompts you to save all unsaved images and then exits. You can also exit ImageJ by clicking on the close button in its window’s title bar.
27  Edit ▶

27.1  Undo [z]
Described in Undo and Redo.

27.2  Cut [x]
Copies the contents of the current image selection to the internal clipboard, filling the selection with the current background color.

SEE ALSO: Edit ▶ Copy to System, Paste Control...

27.3  Copy [c]
Copies the contents of the current image selection to the internal clipboard. If there is no selection, copies the entire active image. The amount of image data copied is shown in the status bar.

SEE ALSO: File ▶ Internal Clipboard, Paste Control...

27.4  Copy to System
Copies the contents of the current image selection to the system clipboard.

SEE ALSO: File ▶ New ▶ System Clipboard [V], Copy [c], Paste Control...

27.5  Paste [v]
Inserts the contents of the internal clipboard (or from the system clipboard if the internal clipboard is empty) into the active image. The pasted image is automatically selected, allowing it to be dragged with the mouse. Click outside the selection to terminate the paste. Select Edit ▶ Undo [z] to abort the paste operation.

SEE ALSO: Paste Control...

27.6  Paste Control...

After pasting, use the Paste Control pop-up menu to control how the image currently being pasted is transferred to the destination image.

Except for Blend and Transparent, the Paste Control transfer modes are the same as those listed in the description of Process ▶ Image Calculator... The Blend mode is the same as the Image Calculator Average mode. In Transparent mode, white/black pixels are transparent and all other pixels are copied unchanged.

SEE ALSO: Image ▶ Overlay ▶ Add Image...

27.7  Clear
Erases the contents of the selection to the current background color. [Backspace] and [Del] keys are shortcuts to this command. With stacks, a dialog is displayed offering the option to clear the selection in all stack images. Clear by pressing [Backspace] to avoid this dialog.

SEE ALSO: Clear Outside, Fill [f], Color Picker Tool

27.8  Clear Outside
Erases the area outside the current image selection to the background color.

SEE ALSO: Clear, Fill [f], Color Picker Tool

27.9  Fill [f]
Fills the current selection with the current foreground color. With stacks, a dialog is displayed offering the option to fill the selection in all stack images. Fill the selection by pressing [F] to avoid this dialog.

SEE ALSO: Clear, Draw [d], Color Picker Tool

27.10  Draw [d]
Outlines the current selection using the current foreground color and line width. The foreground and background colors can also be set using the Edit ▶ Options ▶ Colors... command. Use the Edit ▶ Options ▶ Line Width... command, or double click on the line tool, to change the line width.

With stacks, a dialog is displayed offering the option to draw the selection in all stack images. Draw the selection by pressing [D] to avoid this dialog.

SEE ALSO: Analyze ▶ Label, Color Picker Tool, XVII Embedding Color Annotations in Grayscale Images, XI Drawing Lines Wider Than One-Pixel

XI  Drawing Lines Wider Than One-Pixel
If the line width is an even number, the selection boundary is at the center of the line. If the line width is odd (1, 3, ...) the center of the line drawn is displaced from the selection edge by 1/2 pixel to the bottom right. Thus the line center (the line in case of line width = 1) is inside the selection at the top and left borders, but outside at the bottom and right borders. Rectangular selections (but not polygonal selections or traced selections that happen to be rectangular) are an exception to this rule: For rectangular selections, one-pixel wide outlines are always drawn inside the rectangle. Thicker lines are drawn as for the other selection types.
27.11 Invert [I]

Creates a reversed image, similar to a photographic negative, of the entire image or selection. For 8-bit and RGB images (see Image Types and Formats), Invert always uses min = 0 and max = 255, regardless of the data values. For 16-bit and 32-bit images, the actual minimum and maximum are used (rather than the full range of the pixel type).

See also: Image $\to$ Lookup Tables $\to$ Invert LUT

27.12 Selection

ROI manipulations using Edit $\to$ Selection $\to$ commands. General handling of ROIs and Overlays is described in Selections.

27.12.1 Select All [a]

Creates a rectangular selection that is the same size as the image.

27.12.2 Select None [A]

Deactivates the selection in the active image.

27.12.3 Restore Selection [E]

Restores the previous selection to its original position. A selection is saved when you:

- Delete the selection by clicking outside of it
- Draw a new selection
- De-activate the image containing the selection
- Close the image containing the selection
- Use a command that deletes or modifies the selection

See also: Analyze $\to$ Tools $\to$ ROI Manager

27.12.4 Fit Spline

Fits a cubic spline curve to a polygon or polyline selection (see ROI manipulations).

See also: Straighten...; Interpolate

XII Transferring Selections Between Images

You can transfer a selection from one image to another by activating the image with the selection, activating the destination image, then pressing $\text{Shift} + \text{E}$ (the keyboard shortcut for Edit $\to$ Selection $\to$ Restore Selection [E]). This shortcut can also be used to restore accidentally deleted ROIs. Alternative ways to transfer ROIs across images involve the ROI Manager and the cursor synchronization features provided by Analyze $\to$ Tools $\to$ Synchronize Windows.

27.12.5 Fit Circle

Fits a circle to a multipoint (with at least 3 points) or area selection [14] (see ROI manipulations). Composite selections are not supported. With open shapes (lines and points), the fitting algorithm (Newton-based Pratt fit) described in Pratt V, “Direct least-squares fitting of algebraic surfaces”, Computer Graphics, Vol. 21, pp 145–152 (1987) is used. With closed shapes, the command creates a circle with the same area and centroid of the selection.

27.12.6 Fit Ellipse

Replaces an area selection with the best fit ellipse (see ROI manipulations). The ellipse will have the same area, orientation and centroid as the original selection. The same fitting algorithm is used to measure the major and minor axis lengths and angle when Fit Ellipse is selected in Analyze $\to$ Set Measurements...

See also: DrawEllipse macro

27.12.7 Interpolate

Converts the active selection into a sub-pixel resolution ROI of floating-point coordinates spaced \textit{interval} pixels apart. If Smooth is checked, traced and freehand selections (see Area Selection Tools) are first smoothed using a 3-point running average. Refer to Selections With Sub-pixel Coordinates for more details.

See also: Floating point selections, Fit Spline, Edit $\to$ Options $\to$ Profile Plot Options

27.12.8 Convex Hull

Replaces a polygon of freehand selection with its convex hull (see ROI manipulations), determined by the gift wrap algorithm. The convex hull can be thought of as a rubber band wrapped tightly around the points that define the selection.

See also: Fit Ellipse, \texttt{ConvexitySolidarity} macro, Convex\_Hull\_Plus plugin

27.12.9 Make Inverse

Creates an inverse selection (see ROI manipulations). What is ‘inside’ the selection will be ‘outside’, and vice versa.
27.12.10 Create Mask
Creates a new 8-bit image called ‘Mask’ whose pixels have a value of 255 inside the selection and 0 outside (see ROI manipulations). By default, this image has an inverting LUT, so black is 255 and white is 0 unless Black Background in Process>Binary>Options... is checked.

SEE ALSO: Process>Binary>Convert to Mask, XX Creating Binary Masks, XXII Interpreting Binary Images

27.12.11 Create Selection
Creates a selection from a thresholded image or a binary mask [15].

27.12.12 Properties... [y]
Opens a dialog box that allows the user to assign a contour color (Stroke color) and a contour width (Width) to the active selection or a filling color. Note that selections can be either filled or contoured, but not both. The default selection colors (black, blue, cyan, green, magenta, orange, red, white, yellow) can be typed textually. Any other color must be typed using hex notation (see XIX Hexadecimal Color Values).

Set Stroke width to 0 to have selections drawn with a width of one pixel regardless of the image magnification (see XVIII Working with Zoomed Canvases).

With Text Tool selections, it is also possible to specify the font size and text alignment. Choose List coordinates to retrieve a dedicated table of XY coordinates from the active selection at evenly spaced one pixel intervals.

Note that while this command can only be applied to the active selection, the ROI Manager’s Properties... command (Analyze>Tools>ROI Manager...) can be applied to multiple ROIs.

SEE ALSO: Selections, Image>Overlay>Add Selection... [b], File>Import>XY Coordinates..., File>Save As>XY Coordinates...

27.12.13 Rotate...
Rotates the selection (using floating-point coordinates) by the specified number of degrees. A negative number indicates counter-clockwise rotation. This command runs the RotateSelection macro in ij.jar.

SEE ALSO: FlipSelection macro

27.12.14 Enlarge...
Grows an area selection by a specified number of pixels. Enter a negative value to shrink the selection. This command runs the EnlargeSelection macro in ij.jar. ShrinkSelection is a variation of this macro that does not shrink polygonal selections from the edges of the image.

XIII CONVERTING COMPOSITE SELECTIONS
Enter zero in the Edit>Selection>Enlarge... dialog box to convert Composite Selections into polygon selections. Note, however, that the conversion may fail if the composite ROI is composed of more than one piece and/or contains internal holes.

27.12.15 Make Band...
Takes an area selection and creates a band with a thickness of the specified number of pixels (see ROI manipulations). If you imagine the band as a doughnut shape, then the original selection corresponds to the hole (i.e. the band is made by growing out the original selection).

SEE ALSO: MakeSelectionBand, the macro that implemented this command in previous IJ versions.

27.12.16 Specify...
Opens a dialog that allows the user to define a rectangular or elliptical selection. Width and Height are the dimensions of the selection. X Coordinate and Y Coordinate define the position of the selection. Check Oval to create an elliptical selection. If Centered is checked, the selection is positioned so X Coordinate and Y Coordinate define the center of the selection, otherwise they define the upper left corner.

This command is also available through the ROI Manager More>> drop-down menu (see Analyze>Tools>ROI Manager...).

27.12.17 Straighten...
This command straightens a curved object in an image (see Straightening filamentous objects). The curved object must first be outlined using the Segmented Line Selection Tool. Double click on the line tool icon to open the Image>Adjust>Line Width... widget, in order to adjust the width of the line selection. By default, the Straighten... command fits a cubic spline curve to the points that define the line, so it is not necessary to check the Spline Fit checkbox. Note that Straighten... also works with straight line selections. In this case, the object defined by the line selection is rotated to be horizontal.

SEE ALSO: Straighten plugin, Image>Transform

27.12.18 To Bounding Box
Converts a non-rectangular selection to the smallest rectangle that completely contains it.

27.12.19 Line to Area
Converts a line selection to an area (traced) ROI.
Improved

As described in Segmented Line Selection Tool, the points of a polyline selection can be repositioned (dragged), deleted (using Alt-click) or duplicated (using Shift-click). Press Shift E (Edit ⊿ Selection ⊿ Restore Selection [E]) to restore accidentally deleted lines.

SEE ALSO: Composite Selections, Line to Area

27.12.21 Image to Selection...

Creates an image selection (ImageROI). Image selections are Overlays that can be moved around the canvas (see Image ⊿ Overlay ⊿ Add Image...). Once created, opacity of the blended image can be re-adjusted at any time using Edit ⊿ Selection ⊿ Properties... [y]. Use Shift E (Edit ⊿ Selection ⊿ Restore Selection [E]) to recover the blending image after clicking outside its limits. Use Shift F (Flatten [F]) to finally embed the imageROI.

Note that image selections behave only partially as regular selections (e.g., can be added to the ROI Manager list, can be moved beyond image boundaries but cannot be resized or rotated). However they are stored in the TIFF header and can be saved and restored when saving images in TIFF format.

SEE ALSO: Paste Control... (Blend transfer mode), Image ⊿ Stacks ⊿ Tools ⊿ Insert... , ROI Manager...

27.12.22 Add to Manager [r]

Adds the current selection to the ROI Manager (Analyze ⊿ Tools ⊿ ROI Manager...). If there is no selection the ROI Manager is opened.

27.13 Options>

Use commands in this submenu to change various ImageJ user preference settings.

27.13.1 Line Width...

Displays a dialog box that allows to change the line width (in pixels) of line selections (see Line Selection Tools) and concomitantly the lines generated by the Edit ⊿ Draw [d] command. This legacy command has been superseded by the Image ⊿ Adjust ⊿ Line Width... widget, but required since the latter is not recordable (see Plugins ⊿ Macros ⊿ Record...).

27.13.2 Input/Output...

JPEG quality (0–100) Specifies the compression level used by File ⊿ Save As ⊿ Jpeg... . Requesting a higher degree of compression (a lower value) will result in smaller files, but poorer image quality. Note that lossy JPEG compression creates serious artifacts, see II Image Types: Lossy Compression and Metadata.

GIF and PNG transparent index Specifies the transparent color used for images saved in GIF and PNG formats. Use -1 for “none”. Note that PNG and GIF transparency only works with 8-bit images.

File extension for tables Sets the default extension to be used when saving Results Tables. Files with .txt and .xls extensions are saved in tab-delimited format and files with .csv extensions are saved in comma-delimited format.

Use JFileChooser to open/save Enables versions of File ⊿ Open and File ⊿ Save As that use the Java Swing’s JFileChooser instead of the native OS dialogs. The main advantage of JFileChooser is the ability to open multiple files by Shift-clicking to select multiple contiguous files and control-clicking to select more than one individual file. On the other hand, it is slower, uses more memory, and does not behave like the file open and save dialogs used in other applications. It requires Java 2, which is included with the Linux and Windows distributions of ImageJ and is built into Mac OS X.

Save TIFF and raw in intel byte order Specifies the byte order used when saving 16-bit and 32-bit images using File ⊿ Save As ⊿ Raw Data... , or File ⊿ Save As ⊿ Image Sequence... when Raw is chosen as the format. Check this option to export images using the order used by Intel ×86 based processors (little-endian). This Wikipedia article has more information.

Results Table Options Specifies if column headers and row numbers should be saved or copied from ImageJ tables such as the Results and Summarize windows (see Results Table).
27.13.3 Fonts...

Opens a small widget with three pop-up menus for specifying the typeface, size, style and antialiasing (Smooth checkbox) of the font used by the Text Tool and Image>Stacks>Label... It is also possible to adjust the horizontal text alignment using the style drop-down menu: Left (the default), Right, or Centered. The widget is more easily opened by double clicking on the Text Tool.

27.13.4 Profile Plot Options...

Use this dialog to control how plots generated by ImageJ are displayed (Image>Stacks>Plot Z-axis Profile, Analyze>Plot Profile [k], Analyze>Calibrate..., Analyze>Tools>Curve Fitting..., Multi Plot [Analyzer>Tools>ROI Manager...], etc.).

**Plot Width and Plot Height** Specify the length (in pixels) of the X-axis (Plot Width) and Y-axis (Plot Height).

**Fixed y-axis Scale** If checked, the Y-axis range is fixed and the specified Minimum Y and Maximum Y values are used, otherwise, plots are scaled based on the minimum and maximum gray values.

**Do not Save x–values** If checked, ‘List’, ‘Save...’ and ‘Copy...’ buttons will appear in profile plot windows.

**Auto–close** If checked, profile plot windows will be automatically closed when ‘List’, ‘Save’ and ‘Copy’ are clicked on.

**Vertical Profile** If checked, row average plots of rectangular areas (or line selections wider than 1 pixel) will be generated instead of the default column average plots. Note that evoking Plot Profile [k] with ALT [S] will generate vertical profiles.

**List values** If checked, the list of values will be automatically opened. If Auto–close is also checked, the plot is closed and only the list of values remains open.

**Interpolate line profiles** If checked, Analyze>Plot Profile [k] will use bilinear interpolation to retrieve intensity values along non-straight line selections.

**Draw grid lines** If checked, gray grid lines will be drawn in the plot.

**Sub-pixel resolution** If checked, line selections created on zoomed images will use floating-point coordinates, resulting in smoother curves (see Selections With Sub-pixel Coordinates and Edit>Selection>Interpolate).


27.13.5 Rounded Rect Tool...

See Rounded Rectangular Selection Tool.

27.13.6 Arrow Tool...

See Arrow Tool.

27.13.7 Point Tool...

See Point Tool.

27.13.8 Wand Tool...

See Wand Tool.

27.13.9 Colors...

Displays a dialog box that allows you to set Foreground, Background and Selection color. As mentioned earlier, the selection color is highlighted in the Point Tool and Wand Tool icons. Drawing colors are displayed in the Color Picker Tool (foreground and background colors) and drawing tools such as the Arrow, Brush, Flood Filler and Pencil (foreground color only).

**See also:** Image>Color>Color Picker... [K], Using a Keyboard Shortcut to Change Selection Color

(7) Using a Keyboard Shortcut to Change Selection Color

```java
/* This macro loops through the all the possible selection colors using ↓ “q” as a keyboard shortcut */
var cidx;
macro "Change Selection Color [q]" {
  color= neuart("red", "green", "blue","magenta", "cyan", "yellow", ↔ "orange", "black", "white");
  run("Colors...", "selection"+ color[cidx++]);
  if (cidx=color.length) cidx= 0;
}
```

27.13.10 Appearance...

This dialog contains options that control how images are displayed, an option to display better looking toolbar icons, and an option to set the menu font size.

**Interpolate zoomed images** Uses interpolation instead of pixel replication when displaying zoomed images.

**Open Images at 100%** Newly open images are displayed using 100% magnification (1 image pixel = 1 screen pixel).
**Black Canvas** Causes the image canvas (white by default) to be rendered in black. This is useful when looking at X-ray images in order to avoid high contrasting intensities at the image edges.

**No image border** Displays images without the default one pixel wide black border.

**Use inverting lookup table** Causes newly opened 8-bit images to have inverted pixel values, where white = 0 and black = 255. This is done by both inverting the pixel values and inverting the LUT. Use the Image > Lookup Tables > Invert LUT command to invert an image without changing the pixel values.

**Double Buffer Selections** Reduces flicker when working with complex selections but it also increases memory usage and slows screen updates. It is not needed on Mac OS X, which has built in double buffering.

**Antialiased tool icons** Smooths and darkens the tool icons in the Main ImageJ window. This option is enabled by default on all operating systems. On Windows XP, enable Clear Type sub-pixel anti-aliasing to improve the quality of text in menus.

**Menu font size** Specifies the size of the ImageJ window menu font. Use a size of 0 (zero) to use Java’s default menu font size. Changing the font size requires the restarting of ImageJ. This option is ignored on Mac OS X.


**27.13.11 Conversions...**

Use this dialog to set options that control how images are converted from one type to another.

**Scale When Converting** ImageJ will scale from min–max to 0–255 when converting from 16-bit or 32-bit to 8-bit. Note that Scale When Converting is always checked after ImageJ is restarted.

**Weighted RGB Conversions (0.30, 0.59, 0.11)** When checked, the formula gray = 0.299×red + 0.587×green + 0.114×blue is used to convert RGB images to grayscale. If it is not checked, the formula gray = (red + green + blue)/3 is used. The default weighting factors (0.299, 0.587, 0.114), which are based on human perception, are the ones used to convert from RGB to YUV, the color encoding system used for analog television. The weighting factors can be modified using the setRGBWeights() macro function.

**27.13.12 Memory & Threads...**

Use this dialog to specify the maximum amount of memory available to ImageJ and the number of threads used by filters when processing stacks. Java applications such as ImageJ will only use the memory allocated to them (typically 640 MB) but this dialog allows the user to allocate more than the default.

**Maximum memory** 64-bit OS and a 64-bit version of Java are required to use more than ≈1700 MB of memory. Windows users must be running a 64-bit version of Windows and must install a 64-bit version of Java. Mac users must be running OS X 10.5 or later and may need to use the Java Preferences utility (in /Applications/Utilities/Java) to select a 64-bit version of Java. They may also need to switch to the ImageJ64 application. Linux users need to be running 64-bit versions of Linux and Java. The title of the Memory & Threads dialog box changes to Memory (64-bit) when ImageJ is running on a properly configured 64-bit system.

**Parallel threads for stacks** Determines the number of parallel threads used by commands in the Process > Filters... and the Process > Math... submenus when processing stacks. The default value is the number of available processors.

**Keep multiple undo buffers** If checked, the undo buffer will be preserved when switching images. Edit > Undo [z] remains restricted to the most recent operation, but is available for each opened image, as long as the buffer allows it. If Keep multiple undo buffers is unchecked, the undo buffer is reset every time the active (frontmost) image changes.

**Run garbage collector on status bar click** If checked, forces the Java garbage collector to run every time the user clicks on the Status bar, which may help to reclaim unused memory (see also Plugins > Utilities > Monitor Memory...).


See also: FAQs on the ImageJ wikipage

**27.13.13 Proxy Settings...**

Use this dialog to modify the proxy settings of the Java Virtual Machine. This may be required for ImageJ to connect to the internet in certain machines running behind HTTP proxies. For example, proxy settings may be required to update ImageJ using the Help > Update ImageJ... command or to open the images in the File > Open Samples... submenu.

To use the system proxy settings enable the Or use system proxy settings option (this will set the java.net.useSystemProxies property to true). To configure your proxy settings manually
specify the address of the HTTP proxy in Proxy server and the port the proxy listens on (normally 8080) in Port. Settings will be saved in the ImageJ preferences file (IJPrefs.txt).

27.13.14 Compiler...

Displays a dialog box with options for the Plugins>Compile and Run... command.

**Target** Specifies the Java version of the class files created by Plugins>Compile and Run... Plugins compiled with a Target of 1.6 will not run on earlier versions of Java. A Target of 1.4 should be used to create plugins capable of running on all versions of ImageJ. Target cannot be set higher than the version of Java ImageJ is currently running on.

**Generate Debugging Info (javac -g)** If checked, information needed by Java debuggers will be included in the class files.


27.13.15 DICOM...

This dialog sets options related to the handling of DICOM images. Namely, if ImageJ should open DICOM images as 32-bit float, if voxel depth should be calculated (based on the distance between the first and last slice) and if coronal/transverse sections should be mirrored when using the Image>Stacks>Orthogonal Views [H] command. With IJ 1.45, the DICOM reader applies the Rescale Slope value when Open as 32-bit float is enabled and tag 0028,1053 is not 1.0.

**See also:** Image Types and Formats

27.13.16 Misc...

Displays a dialog box for configuring several (advanced) settings that do not fit elsewhere in the ImageJ interface.

**Divide by zero value** Specifies the value used when Process>Image Calculator... detects a divide by zero while dividing one 32-bit real image by another. The default is infinity. In addition to numeric values, "infinity" (positive or negative infinity), "max" (largest positive value) and "NaN" (Not-a-Number) can be entered as the Divide by zero value.

**Use pointer cursor** If checked, ImageJ will use an arrow cursor instead of the default crosshair.

As mentioned in Settings and Preferences, IJPrefs.txt holds all the settings and preferences of ImageJ and is stored in ~/Library/Preferences/ on Mac OS X, in ~/.imagej/ on Linux and Windows (with ~ referring to the user’s home directory). Several macros and plugins also write parameters to this file.

**See also:** (3) Ensuring Specific Settings at Launch, Plugins>Utilities>Reset...
28 Image ▶

28.1 Type ▶

Use this submenu to determine the type of the active image or to convert it to another type. An attempt to perform an unsupported conversion causes a dialog box to be displayed that lists the possible conversions.

Supported conversions in ImageJ (Image ▶ Type ▶ submenu). Note that ImageJ supports many more types of image data.

<table>
<thead>
<tr>
<th>▶From To ▶</th>
<th>8-bit</th>
<th>16-bit</th>
<th>32-bit</th>
<th>8-bit color</th>
<th>RGB color</th>
<th>RGB stack</th>
<th>HSB stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>8-bit</td>
<td>...</td>
<td>I, S</td>
<td>I, S</td>
<td>I, S</td>
<td>I, S</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16-bit</td>
<td>I, S</td>
<td>...</td>
<td>I, S</td>
<td>I, S</td>
<td>I, S</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32-bit</td>
<td>I, S</td>
<td>I, S</td>
<td>...</td>
<td>I, S</td>
<td>I, S</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8-bit color</td>
<td>I, S</td>
<td>...</td>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RGB color</td>
<td>I, S</td>
<td>I, S</td>
<td>...</td>
<td>I, S</td>
<td>I, S</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RGB stack</td>
<td>I</td>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HSB stack</td>
<td>I</td>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

I: Single images only; S: Stacks

8-bit Converts to 8-bit grayscale. ImageJ converts 16-bit and 32-bit images to 8-bit by linearly scaling from min-max to 0–255, where min and max are the two values displayed in the Image ▶ Adjust ▶ Brightness/Contrast... [C]. Image ▶ Show Info... [I] displays these two values as Display range. Note that this scaling is not done if Scale When Converting is not checked in Edit ▶ Options ▶ Conversions... RGB images are converted to grayscale using the formula gray = (red + green + blue)/3 or gray = 0.299 × red + 0.587 × green + 0.114 × blue if Weighted RGB Conversions is checked in Edit ▶ Options ▶ Conversions...

16-bit Converts to unsigned 16-bit grayscale.

32-bit Converts to signed 32-bit floating-point grayscale.

8-bit Color Converts to 8-bit indexed color using Heckbert’s median-cut color quantization algorithm. A dialog box allows the number of colors (2–256) to be specified. The active image must be RGB color.

RGB Color Converts to 32-bit RGB color.

HSB Stack Converts to a 3-slice (hue, saturation and brightness) stack. The active image must be RGB color.

28.2 Adjust ▶

This submenu contains commands that adjust brightness/contrast, threshold levels and image size.

XIV Applying Auto Brightness/Contrast to Entire Stacks

The Process ▶ Enhance Contrast... command can be used to adjust the brightness and contrast of each slice in a stack, according to either the optimal for each individual slice (if Use Stack Histogram is unchecked) or the overall stack (by ticking Use Stack Histogram). The default behavior of the B&C tool (Image ▶ Adjust ▶ Brightness/Contrast... [C]) is to use the overall stack histogram.

28.2.1 Brightness/Contrast... [C]

Use this tool to interactively alter the brightness and contrast of the active image. With 8-bit images, brightness and contrast are changed by updating the image’s lookup table (LUT), so pixel values are unchanged. With 16-bit and 32-bit images, the display is updated by changing the mapping from pixel values to 8-bit display values, so pixel pixel values are also unchanged. Brightness and contrast of RGB images are changed by modifying the pixel values.

Histogram The line graph at the top of the window, which is superimposed on the image’s histogram, shows how pixel values are mapped to 8-bit (0–255) display values. The two numbers under the plot are the minimum and maximum displayed pixel values. These two values define the display range, or “window”. ImageJ displays images by linearly mapping pixel values in the display range to display values in the range 0–255. Pixels with a value less than the minimum are displayed as black and those with a value greater than the maximum are displayed as white.

Minimum and Maximum sliders Control the lower and upper limits of the display range. Holding down Shift will simultaneously adjust all channels of a composite image (e.g., File ▶ Open Samples ▶ HeLa Cells (1.3 M, 48-bit RGB)).

Brightness slider Increases or decreases image brightness by moving the display range. Holding down Shift will simultaneously adjust all channels of a composite image.

Contrast slider Increases or decreases contrast by varying the width of the display range. The narrower the display range, the higher the contrast. Holding down Shift will simultaneously adjust all channels of a composite image.

Auto ImageJ will automatically optimize brightness and contrast based on an analysis of the image’s histogram. Create a selection, and the entire image will be optimized based on an analysis of the selection. The optimization is done by allowing a small percentage of pixels in the image to become saturated (displayed as black or white). Each additional click on Auto increases the number of saturated pixels and thus the amount of optimization. A run (“Enhance Contrast”, *saturated=0.35”) macro call is generated if the command recorder (Plugin ▶ Macro ▶ Record...) is running.

Reset Restores the original brightness and contrast settings. The display range is set to the full pixel value range of the image. A resetMinAndMax() macro call is generated if the command recorder is running. Holding down Shift restores original settings in all channels of a composite image.

Set Allows to enter the minimum and maximum display range values in a dialog box. A setMinAndMax() macro call is generated if the command recorder is running.
A 16-bit image consists of 65536 possible gray levels. Most of times, however, the relevant image information is contained only within a narrow range of the grayscale. This is the case, e.g., in low light microscopy, in which signal is restricted to the lower end of the grayscale. The Set Display Range dialog allows you to choose how to scale the range of gray levels of 16-bit images.

**Automatic** Automatically selects the best range given the intensity values of the image based on the percentage of the total number of pixel values from the lowest to highest pixel value.

- **8-bit (0–255)** Gray level range of 0–255.
- **10-bit (0–1023)** Gray level range of 0–1023.
- **12-bit (0–4095)** Gray level range of 0–4095.
- **15-bit (0–32767)** Gray level range of 0–32767.
- **16-bit (0–65535)** Gray level range of 0–65535.

Check **Propagate to all open images** to apply these values to the rest of the images currently open. With multi-channel images, the option to propagate the specified range to the remaining channels is also available.

**Apply** Applies the current display range mapping function to the pixel data. If there is a selection, only pixels within the selection are modified. This option currently only works with 8-bit images, 8-bit stacks and RGB stacks. This is the only B&C option that alters the pixel data of non-RGB images.

**See also:** Window/Level..., Enhance Contrast..., Color Balance..., XIV Applying Auto Brightness/Contrast to Entire Stacks, XVI Brightness/Contrast of High Bit–Depth Images, XV Display Range of DICOM Images

### 28.2.2 Window/Level...

This command and Brightness/Contrast... [C] (B&C) are redundant, but Window/Level... (W&L) behaves in a manner closer to that implemented on medical image terminals by interactively adjusting the Window – range of minimum and maximum (Contrast) – and Level – position of that range in the grayscale intensity space (Brightness).

If the B&C window is opened, it will be closed and the W&L window will be opened at the same location.

### 28.2.3 Color Balance...

This panel makes adjustments to the brightness and contrast of a single color of a standard RGB image (8-bit per color channel).

For multi-channels Stacks and Hyperstacks (Color Composite Images) it adjusts each of the color channels independently. Use the drop-down menu to specify which color / channel will be adjusted (the histogram is drawn for the selected channel).

**Maximum** and **Minimum** sliders, Auto, Set and Apply work as described for Image→Adjust→Brightness/Contrast... [C]. Similarly to the Window/Level... tool, if the B&C window is opened, it will be closed and the Color window will be opened at the same location.

**NB:** When switching from one color to another, the changes made to one color will be lost unless Apply is clicked before. Also, note that for 48-bit color images that load as a stack, Brightness/Contrast... [C] works on single stack slices, i.e., colors, and the color settings of the Color panel are ignored.

**See also:** Brightness/Contrast... [C], Color→submenu

### 28.2.4 Threshold... [T]

Use this tool to automatically or interactively set lower and upper threshold values, segmenting grayscale images into features of interest and background. Use Analyze→Measure... [m] (with Limitto Threshold in Analyze→Set Measurements... checked) to measure the aggregate of the...
XVI  Brightness/Contrast of High Bit–Depth Images

When displayed, the intensity of each pixel that is written in the image file is converted into the grayness of that pixel on the screen. How these intensities are interpreted is specified by the image type. From the ImageJ website:

16-bit and 32-bit grayscale images are not directly displayable on computer monitors, which typically can show only display 256 shades of gray. Therefore, the data are mapped to 8-bit by windowing. The window defines the range of gray values that are displayed: values below the window are made black, while values above the window are white. The window is defined by minimum and maximum values that can be modified using Image > Adjust > Brightness/Contrast... [C].

It may happen that the initial windowing performed by ImageJ on these high bit–depth (or HDR) images is suboptimal. Please note that windowing does not affect image data (cf. the HDRExplorerTool).

Auto-threshold methods

- Default
- Huang
- IsoData
- Li
- MaxEntropy
- Mean
- MinEntropy
- Minimum
- Maximum
- Otsu
- Percentile
- RenyiEntropy
- ShannonEntropy
- Triangle
- Yen

Image > Adjust > Threshold... [T] (ImageJ 1.45m).

selected features. Use Analyze > Analyze Particles... to measure features individually. Use the Wand Tool to outline a single feature.

Upper slider  Adjusts the minimum threshold value. Hold Shift while adjusting the minimum to move a fixed-width thresholding window across the range of gray values.

Lower slider  Adjusts the maximum threshold value.

Method  Allows any of the 16 different automatic thresholding methods to be selected [18]. These methods are described on Fiji’s Auto Threshold website. The Default method is the modified IsoData algorithm used by ImageJ 1.41 and earlier. Note that these are global thresholding methods that typically cannot deal with unevenly illuminated images (such as in brightfield microscopy). In these cases, local algorithms are more appropriate, by allowing the threshold to smoothly vary across the image. These are implemented by the Auto Local Threshold plugin, pre-installed in Fiji.

Display  Selects one of three display modes:

- Red  Displays the thresholded values in red.
- B&W  Features are displayed in black and background in white. This mode respects the Black background flag set in Process > Binary > Options...
- Over/Under  Displays pixels below the lower threshold value in blue, thresholded pixels in grayscale, and pixels above the upper threshold value in green. These colors can be changed from a macro by calling the ImageProcessor.setOverColor() and setUnderColor() methods (example).

Dark background  To be checked when features are lighter than the background. The state of the checkbox is remembered across restarts.

Stack histogram  If checked, ImageJ will first compute the histogram of the whole stack (or hyperstack) and then compute the threshold based on that histogram. As such, all slices are binarized using the single computed value. If unchecked, the threshold of each slice is computed separately.

Auto  Uses the currently selected thresholding method to automatically set the threshold levels based on an analysis of the histogram of the current image or selection.

Apply  Sets thresholded pixels to black and all other pixels to white. For 32-bit float images Apply will also run Process > Math > NaN Background.

Reset  Disables thresholding and updates the histogram.

Set  New threshold levels can be entered into this dialog box.

See also:  XX Creating Binary Masks, Color Threshold..., Wand Tool, Analyze > Analyze Particles...  

28.2.5  Color Threshold...

Thresholds 24-bit RGB images based on Hue Saturation and Brightness (HSB), Red Green and Blue (RGB), CIE Lab or YUV components. Ranges of the filters can be set manually or based on the pixel value components of a user-defined ROI. This command, implemented in version 1.43l, is an experimental built-in version of the Threshold Colour plugin [28] and is not yet fully integrated into ImageJ.

Segmentation of DAPI stained nuclei using Image > Adjust > Color Threshold...
**Pass**  If checked, values within range are thresholded and displayed (band-pass filter), otherwise, values outside the selected range are thresholded (band-reject filter).

**Thresholding Method**  Allows any of the 16 different automatic thresholding methods to be selected (see Threshold... [T]).

**Threshold Color**  Selects the threshold color: either Red, Black, White or Black & White (see Threshold... [T]).

**Color space**  Selects the color space: HSB, RGB, CIE Lab or YUV (see Color Spaces and Color Separation).

**Dark background**  To be checked when features are lighter than the background (see XXII Interpreting Binary Images). The state of the checkbox is remembered across restarts.

**Original**  Restores the original image and updates the buffer when switching to another image.

**Filtered**  Shows the filtered image. Note that the final thresholded image type is RGB, not 8-bit gray (see Image Types and Formats).

**Select**  Creates a ROI selection based on the current settings. The selection is made according to the settings defined in the Process>Binary>Options... dialog.

**Sample**  (Experimental) Sets the ranges of the filters based on the pixel value components in a user-defined ROI.

**Stack**  Processes the remaining slices of the stack (if any) according to the current settings.

**Macro**  Creates a macro based on the current settings which is sent to the Macro Recorder window (Plugins>Macros>Record...). If the canvas size is increased, the border is filled with 8-bit gray (see Image Types and Formats).

**Help**  Opens the built-in help dialog.

**See also:**  3D Color Inspector/Color Histogram, Threshold... [T], Wand Tool, Analyze>Analyze Particles...

### 28.2.6 Size...

Scales the active image or selection to a specified Width and Height in pixels.

Check **Constrain aspect ratio** and ImageJ will adjust either the Height or the Width to maintain the original aspect ratio. When applicable, other dimensions can also be resized: Depth (images) in stacks, Depth (slices) and Time (frames) in hyperstacks.

Check the **Average when downsizing** checkbox for better results when scaling down images [19]. Two resampling methods are possible: Bilinear and Bicubic interpolation. The implementation of the bicubic method (Catmull-Rom interpolation) is derived from Burger and Burge, 2008 [11].

**See also:**  Canvas Size..., Image>Scale... [E], Image<Transform>Bin...
28.4 Properties... [P]

Use this command to display and set various properties of the current image or stack.

The number of Channels (c), Slices (z) and Frames (t) in the image can be changed as long as the product of c, z, and t is equal to the number of images in the stack.

Unit of Length is a string describing the measuring unit of Pixel Width, Pixel Height and Voxel Depth. These three dimensions are automatically converted if Unit of Length is changed from one of ImageJ’s known unit (‘nm’, ‘µm’ [or ‘µm’ or ‘micron’], ‘mm’, ‘cm’, ‘meter’, ‘km’ or ‘inch’) to another. µ and Å symbols can be typed using [Alt M] and [Alt Shift A], respectively.

With t-series stacks, the Frame Interval (in seconds (reciprocal of the frame rate) can be viewed and set. If the unit is ‘sec’, setting the Frame Interval will also set the frame rate used by Animation Options... [Alt/].

Origin is the reference point 0,0 (always in pixels) of the image coordinate system (see also Invert Y coordinates in Analyze> Set Measurements...).

As explained in XXIII Global Calibrations, check Global to make the current settings global, i.e., applied to all images opened during the current session.

See also: Analyze> Set Scale... , Image> Show Info... [i]

28.5 Color>[P]

This submenu contains commands that deal with color images.

28.5.1 Split Channels

Splits an RGB image (or stack) into three 8-bit grayscale images containing the red, green and blue components of the original. The window names have an appended (red), (green) and (blue).

With composite images and / or hyperstacks (e.g., File> Open Samples> Organ of Corti (2.8M, 4D stack)) this command splits the stack into separate channels.

See also: Merge Channels... [Z]

28.5.2 Merge Channels...

Merges 2–7 images into an RGB image or multi-channel composite image. Select the channel order/color using the C1–C7 dropdown menus. Select ‘None’ to skip a channel.

Create composite If checked, a multi-channel composite image (see Color Composite Images) will be created. If unchecked, an RGB image is created instead. When creating composite images, original LUTs and display ranges are preserved unless Ignore source LUTs is checked. Source LUTs are always ignored when creating RGB images.

Keep source Images If checked, source images will not be disposed.

Ignore source LUTs If checked, LUTs of source images are ignored. In this case, merged channels will adopt the lookup table mentioned besides the channel choice, i.e., red, green, blue, gray, cyan, magenta, yellow. As mentioned, this option is assumed when merging into RGB.

See also: Channels Tool... [Z], Lookup Tables... and IV Replacing Red with Magenta in RGB Images

28.5.3 Channels Tool... [Z]

Alias for Image> Hyperstacks> Channels Tool... [Z].

28.5.4 Stack to RGB

Converts a two or three slice stack into an RGB image, assuming that the slices are in R, G, B order. The stack must be 8-bit or 16-bit grayscale. Also converts composite images (e.g., File> Open Samples> HeLa Cells (1.3M, 48-bit RGB)) into RGB.

28.5.5 Make Composite

Converts in place an RGB image, a 2–7 image stack or a 2–7 channel hyperstack into a composite color image. Use the Channels Tool... [Z] tool ( Shift Z ) to enable and disable the channels of a composite image. Use Brightness/Contrast... [C] ( Shift C ) to adjust the brightness and contrast of the current channel.

28.5.6 Show LUT

Displays a plot of the active image’s lookup table (LUT). The lookup table, or color table, describes the color that is displayed for each of the 256 possible pixel values. For 16 and 32-bit images, the range of displayed pixel values is mapped to 0–255. A bar under the plot displays the color representation of the pixel values. Note that RGB color images do not use a lookup table. Use the List... radio button to export the LUT as a CSV file.

See also: Edit LUT...
28.5.7 Edit LUT…

Opens the ImageJ LUT (Lookup Table) Editor. A lookup table in ImageJ has up to 256 entries. The entry index, and the three values (red, green and blue) associated with it, are displayed in the ImageJ status bar as you move the cursor over the LUT Editor window. Click on an entry to edit the red, green and blue values for that entry using a Color Selector window (cf. Color Picker… [K]).

See also: Show LUT

28.5.8 Color Picker… [K]

The Color Picker [17] enables the user to select foreground and background colors, which affects Edit>Fill [f], Draw [d], Clear and other drawing commands. It displays current foreground and background colors in the selection boxes at the bottom of the window. It has two modes: Foreground and Background. To change modes, click on the desired selection box. Clicking on the Foreground/Background Switcher button sets the current foreground to the background and vice versa. The Black/White Reset button sets the foreground to black and the background to white.

Double clicking on a color brings up the ColorChooser, a widget with three sliders used to specify the RGB values of the foreground or background color. The title of the ColorChooser widget (Foreground Color or Background Color) indicates the current selection mode. To get precise colors, manually change the values in the text boxes. The hex value of the final color is also displayed, offering a convenient way to retrieve custom colors to, e.g., personalize Overlays (see XIX Hexadecimal Color Values).

As mentioned earlier, the Color Picker Tool tool can be used to ‘pick-up’ foreground/background colors from an image canvas. Foreground color can also be changed using the Color dropdown menu in the Options dialog of drawing tools such as Arrow, Brush, Overlay Brush and Pencil tools.

See also: XVII Embedding Color Annotations in Grayscale Images, Draw [d], Fill [f], Clear, Clear Outside, Image Types and Formats, Temporary Activation of a Tool

Color Pickers and Color Choosers (IJ 1.46r). The CP window can be activated using Shift+K (the keyboard shortcut for Image>Color>Color Picker… [K]) or by double clicking on the Color Picker Tool on the ImageJ Toolbar. Color Choosers are evoked by double clicking on a color of the CP window and can be used to retrieve Hexadecimal Color Values. The ‘eye dropper’ is drawn in the current foreground color while the frame around it is drawn in background color. Foreground color is also reflected in drawing tools such as the Arrow, Brush, Flood Filler and Pencil tools.

The color palette is based on HSB (Hue, Saturation and Brightness) color model (see Color Spaces and Color Separation). Hue increases as you go down the palette while saturation and brightness values are split horizontally. The left half of the palette varies only in brightness while the right half varies only in saturation. At the center of the color ramp are enlarged red, green, blue, cyan, magenta, and yellow colors for quick selection. To the left of the color palette is a grayscale ramp that goes from pure black to pure white.
28.6.3 Next Slice [>] 
Displays the slice that follows the currently displayed slice. Holding \( \text{Alt} > \) will skip ten slices forward.

See also: Arrow Keys

28.6.4 Previous Slice [<] 
Displays the slice that precedes the currently displayed slice. Holding \( \text{Alt} < \) will skip ten slices backward.

See also: Arrow Keys

28.6.5 Set Slice... 
Displays a specified slice. The user must enter a slice number greater than or equal to one and less than or equal to the number of slices in the stack.

28.6.6 Images To Stack 
Creates a new stack from images currently displayed in separate windows.

**Method** If images differ in size, a drop-down menu allows to choose a conversion method:

- **Copy (center) and Copy (top-left)** 
  Stack will have the width of the widest open image and the height of the highest open image. Smaller images will then be copied (either to the center or to the upper left corner) of the slice. Borders are filled with pixels that have a value of zero.

- **Scale (smallest) and Scale (largest)** 
  Stack will have the dimensions of the smallest/largest open image. Other images are scaled to the new slice dimensions. Bicubic interpolation is used if Bicubic interpolation is checked (cf. Image Size... and Image Scale...).

**Name** Specifies the title of the stack to be created.

**Title Contains** Enter a string into this field and ImageJ will only convert to stack images whose name contains that string.

**Bicubic Interpolation** If checked, bicubic interpolation (cf. Adjust Size...) will be used if any of the Scale methods was previously chosen.

**Use Title as Labels** If checked, image titles (without extension) will be used as stack labels. As described in Stacks > Tools > Remove Slice Labels, these labels (up to 60 characters) correspond to the image subtitle, the line of information above the image.

**Keep Source Images** If checked, original images are kept.

28.6.7 Stack To Images 
Converts the slices in the current stack to separate image windows.

See also: Stacks > Images To Stack

28.6.8 Make Montage... 
Produces a single grid-image containing the individual images that compose Stacks and 4D Hyperstacks. This can be useful for visual comparisons of a series of images stored in a stack and to create ‘panel figures’ for publication and presentations. A dialog box allows you to specify the magnification level at which the images are copied, and to select the layout of the resulting grid.

**Label Slices** If checked, montage panels are labelled with slice labels. Slice labels (up to 60 characters) correspond to the image subtitle, the line of information above the image. These labels are part of the stack metadata, typically created by File > Import > Image Sequence... or Stacks > Images To Stack. If no slice metadata exists (the setMetadata("Label", string) macro function can be used to customize slice labels) images are labelled with slice numbers. Note that the Stacks > Label... command can be used to draw labels in stack slices. Labels are typeset in sans-serif typeface.

**Use Foreground Color** If checked, borders and labels are drawn in the foreground color and blank areas of the panel are filled with the background color.

See also: Stacks > Tools > Montage to Stack... Stacks > Tools > Remove Slice Labels, RC Montage plugin, Magic Montage — a macro toolset to reorder and manipulate images in the montage (a video tutorial can be found here)
Reconstructs one or more orthogonal slices through the image volume represented by the current stack or hyperstack [21].

The estimated size of the output stack and the amount of available memory are displayed at the bottom of the dialog. Increase Output spacing to reduce the size of the output stack.

A dialog allows you to specify the spacing of the reconstructed slices.

**Output spacing** Determines the number of orthogonal slices that will be reconstructed. Increasing Output spacing reduces the size of the output stack.

**Start at** Determines the image edge (top, left, bottom or right) from which reconstruction starts. Start at is replaced by Slice count if there is a line selection. With lines selections, a stack is created by shifting (by Output spacing) the line down and to the left to generate additional slices for the output stack. In this case, the size of the output stack in determined by Slice count.

**Flip vertically** If checked, each slice in the output stack will be flipped vertically.

**Rotate 90 degrees** If checked, each slice in the output stack will be rotated 90°.

**Avoid interpolation** If checked, no interpolation will be done.


See also: Dynamic Reslice and Radial Reslice plugins

---

Provides an orthogonal view display of the current stack or hyperstack [22]. E.g., if a stack displays sagittal sections, coronal (YZ projection image) and transverse (XZ projection image) will be displayed through the data-set.

The two extra planar views are displayed in ‘sticky’ panels next to original image and can be toggled using **Shift H**, the command shortcut.

The intersection point of the three views follows the location of the mouse click and can be controlled by clicking and dragging in either the XY, XZ or YZ view. XY and XZ coordinates are displayed in the title of the projection panels. The mouse wheel changes the screen plane in all three views.

Voxel dimensions can be adjusted in **Image Properties**... [P].

---

Projects an image stack along the axis perpendicular to image plane (the so-called z axis) [21]. With hyperstacks, the projection is performed on the active time frame, or for all time points if **All Time Frames** is checked. Five different projection types are supported. The preferred projection method is stored in the preferences file.

**Average Intensity** projection outputs an image wherein each pixel stores average intensity over all images in stack at corresponding pixel location.

**Maximum Intensity** projection (MIP) creates an output image each of whose pixels contains the maximum value over all images in the stack at the particular pixel location.

**Sum Slices** projection creates a real image that is the sum of the slices in the stack.

**Standard Deviation** projection creates a real image containing the standard deviation of the slices.

**Median** projection outputs an image wherein each pixel stores median intensity over all images in stack at corresponding pixel location.

See also: Grouped Z Project... , 3D Project... , Plot Z-axis Profile...
of various visualization parameters determine how both surface and interior structures will appear.

**Projection Method** Select *Nearest Point* projection to produce an image of the surfaces visible from the current viewing angle. At each point in the projection plane, a ray passes normal to the plane through the volume. The value of the nearest non-transparent point which the ray encounters is stored in the projection image. *Brightest Point* projection examines points along the rays, projecting the brightest point encountered along each ray. This will display the brightest objects, such as bone in a CT (computed tomographic) study. *Mean Value* projection, a modification of brightest-point projection, sums the values of all transparent points along each ray and projects their mean value. It produces images with softer edges and lower contrast, but can be useful when attempting to visualize objects contained within a structure of greater brightness (e.g. a skull).

**Slice Spacing** The interval, in pixels, between the slices that make up the volume. ImageJ projects the volume onto the viewing plane at each *Rotation Angle Increment*, beginning with the volume rotated by *Initial Angle* and ending once the volume has been rotated by *Total Rotation*.

**Lower / Upper Transparency Bound** Determine the transparency of structures in the volume. Projection calculations disregard points having values less than the lower threshold or greater than the upper threshold. Setting these thresholds permits making background points (those not belonging to any structure) invisible. By setting appropriate thresholds, you can strip away layers having reasonably uniform and unique intensity values and highlight (or make invisible) inner structures. Note that you can also use *Image > Adjust > Threshold...* to set the transparency bounds.

**Opacity** Can be used to reveal hidden spatial relationships, especially on overlapping objects of different colors and dimensions. The (surface) *Opacity* parameter permits the display of weighted combinations of nearest-point projection with either of the other two methods, often giving the observer the ability to view inner structures through translucent outer surfaces. To enable this feature, set *Opacity* to a value greater than zero and select either *Mean Value* or *Brightest Point* projection.

**Surface / Interior Depth-Cueing** Depth cues can contribute to the three-dimensional quality of projection images by giving perspective to projected structures. The depth-cueing parameters determine whether projected points originating near the viewer appear brighter, while points further away are dimmed linearly with distance. The trade-off for this increased realism is that data points shown in a depth-cued image no longer possess increased realism from the current foreground color (cf. *Image > Colors > Color Picker...*).

The initial *X, Y location*, and *Font size* of the label are based on the existing rectangular selection, if any. Slices outside the *Range* are not affected.

**Starting value and Interval** Specify the first value and the numeric steps to be applied. Note that with timestamps, metric time values must be used, e.g., an *Interval* of 3600 will create 1 hour increments.

**Text** The string to be drawn after each number when the *Format* chosen is either 0 or *Text* (label without numeric sequence).

**Use overlay** If checked, labels will be created using the typeface and style specified in the *Fonts...* widget. If unchecked, labels are typeset using ImageJ’s default font: sans-serif typeface.

28.6.15 Tools

28.6.15.1 Combine...

Combines two stacks \([\text{Width} \times \text{Height} \times \text{Depth}]\) \((W_1 \times H_1 \times D_1\) and \(W_2 \times H_2 \times D_2)\) to create a new \(W_1 + W_2 \times \max(H_1, H_2) \times \max(D_1, D_2)\) stack. E.g., a 256\times 256\times 40 and a 256\times 256\times 30 stack would be combined to create one 512\times 256\times 40 stack.

If Combine vertically is enabled, creates a new \(\max(W_1 + W_2) \times (H_1 + H_2) \times \max(D_1, D_2)\) stack.

Unused areas in the combined stack are filled with background color (cf. Color Picker... [K]).

See also: Concatenate...

28.6.15.2 Concatenate...

Concatenates multiple images or stacks. Images with mismatching type (see Image Types and Formats) and dimensions are omitted [22]. Stacks with the same number of slices can be concatenated as a 4D Hyperstacks, if Open as 4D image is checked.

In this case, chosen stacks will be appended as time-points.

See also: Combine...

28.6.15.3 Reduce...

Reduces the size of stacks and hyperstacks by the specified Reduction Factor. E.g., For a 30 slices stack and a Reduction Factor of 2, the reduced stack will be composed of 15 slices with every second slice being removed. Virtual stacks/hyperstacks are supported.

With Hyperstacks, the default reduction is performed in the T-Dimension, but a choice is available to Reduce in Z-Dimension instead.

See also: Hyperstacks Repeat Dimensionality...

28.6.15.4 Reverse

Alias for the Image Transform Flip Z command.

28.6.15.5 Insert...

Inserts a Source image into a Destination image at the specified X and Y Location (pixel coordinates). Source and Destination can be single images or stacks but must be of the same type (see Image Types and Formats). The Destination image will be permanently modified once Source has been inserted. Note that when Source is a single image, Edit Selection Image to Selection... can be used (together with Image Overlay Add image...) to create image selections (ImageROIs), a more convenient way of blending two open images.

See also: Image Type, Image Stacks Label...

28.6.15.6 Montage to Stack...

Converts a montage image to an image stack based on the specified number of rows and columns, taking into account a Border width. This is the opposite of what the Image Stacks Make Montage... command does.

See also: Demontager plugin

28.6.15.7 Make Substack...

Extracts selected images from the active stack copying them to a new stack in the order of listing or ranging [25]. Extracted slices will be removed from the source stack if Deleted slices from original stack is checked. Currently, it does not work with hyperstacks and takes one of three types of input:

1. A range of images. E.g.: 2–14 [extract slices 2 through 14]
2. A range of images with increment, which can be used to de-interleave slices. E.g.: 2–14–2 [extract slices 2 and 14 and every second slice in between]
3. A list of images. E.g.: 2, 4, 7, 9, 14

See also: Image Duplicate... [D]

28.6.15.8 Grouped Z Project...

Creates a substack of Stack size/Group size slices with each slice being the result of a Z Projection performed over the range of Group size.
28.7 Hyperstacks

This submenu hosts commands specifically related Hyperstacks, images that have four (4D) or five (5D) dimensions. General operations related to Stacks are listed in the \( \text{Image}\to \text{Stacks} \) submenu.

28.7.1 New Hyperstack...

Creates a new hyperstack. Hyperstacks have Width, Height, Channels (c dimension), Slices (z dimension) and time Frames (t dimension).

Image Type (see \( \text{Image}\to \text{Type} \) and Image Types and Formats) and Display Mode (see Channels Tool... [Z]) can be specified.

Checking Label Images will draw the channel number, slice number and frame number on each image in the hyperstack.

See also: File\to New\to Hyperstack... (an alias of this command) and File\to New\to Image... [n]

28.7.2 Stack to Hyperstack...

Converts a stack into a hyperstack. RGB stacks are converted into 3 channel hyperstacks.

Order is the order of the channels (c), slices (z) and frames (t) within the stack. ImageD hyperstacks are always in czt order. Stacks not in czt order will be shuffled to be in czt order. The channel Display Mode can be Composite, Color or Grayscale (cf. Channels Tool... [Z]).

See also: Hyperstack to Stack

28.7.3 Hyperstack to Stack

Converts a hyperstack into a stack (in czt order).

See also: Stack to Hyperstack...

28.7.4 Reduce Dimensionality...

This command [26] reduces the dimensionality of an hyperstack by creating a new hyperstack with, for example, all the channels and time points at a given z position or all the z slices for the current channel and time point.

See also: File\to Save As\to Gif... , AVI...
Uncheck **Channels** \((n)\) to delete all but the current channel, **Slices** \((n)\) to delete all but the current z slice and **Frames** \((n)\) to delete all but the current time point. Check **Keep Source** and the original stack will not be deleted.

The expected dimensions and size of the reduced stack are displayed in the dialog.

**SEE ALSO:** Hyperstack to Stack

### 28.7.5 Channels Tool... \([Z]\)

Opens the Channels... widget, or brings it to the front if it is already open. \(\text{Shift} \ Z\) is the keyboard shortcut for this command. This tool allows to select the **Display mode** of composite images. In addition, several commands hosted in the **Image ⊲ Color ⊲ Channels Tool...** tool can easily be accessed through the More≫ drop-down menu. The same drop-down menu also provides a convenient list of primary colors (additive: red, green and blue, subtractive: cyan, magenta, yellow) that can be used to pseudocolor Color Composite Images.

Channel manipulations in Color Composite Images using the Image ⊲ Color ⊲ Channels Tool... [Z] tool.

### 28.8 Crop [X]

Crops the image or stack based on the current rectangular selection.

### 28.9 Duplicate... \([D]\)

Creates a new window containing a copy of the active image or rectangular selection. For stacks and hyperstacks it is possible to specify the range of **Channels** \((c)\), **Slices** \((z)\) and **Frames** \((t)\) to be duplicated.

With single images, hold [Alt] to skip the dialog box.

**SEE ALSO:** Rename...

---

**28.10 Rename...**

Renames the active image.

**SEE ALSO:** Duplicate... \([D]\)

**28.11 Scale... \([E]\)**

Resizes the image or current area selection by scale factors entered into a dialog box. As with **Image ⊲ Size...**, two resampling methods are possible: **Bilinear** and **Bicubic** interpolation.

For the best looking results, particularly with graphics and text, use integer scale factors (2, 3, 5, etc.) and check **Average when downsizing** with scale factors less than 1.0 \([19]\). Also, when downsizing, smoothing the source image prior to scaling may produce better looking results.

Scaled image/selection are copied to a new image named **Title** if **Create new window** is checked. If scaling a selection that will not be copied to a new image check **Fill with Background Color** to fill with the background color instead of zero.

Entire stacks (or hyperstacks in the Z Dimension) will be scaled if **Process entire stack** is checked.

**SEE ALSO:** Image ⊲ Adjust ⊲ Size..., Image ⊲ Transform ⊲ Bin..., Image ⊲ Stacks ⊲ Tools ⊲ Grouped Z Project...

### 28.12 Transform>

This submenu contains commands that perform geometrical image transformation on the active image or stack.

#### 28.12.1 Flip Horizontally

Replaces the image or selection with a x-mirror image of the original.

#### 28.12.2 Flip Vertically

Turns the image or selection upside down (y-mirror).

#### 28.12.3 Flip Z

Reverses the order of the slices in a stack (x-mirror).

#### 28.12.4 Rotate 90 Degrees Right

Rotates the entire image or stack clockwise 90°.
28.12.5 Rotate 90 Degrees Left

Rotates the entire image or stack counter-clockwise 90°.

With 8-bit and RGB images, check Fill with Background Color to fill with the background color instead of zero (cf. Color Picker... [K]). Check Enlarge to Fit Result and the image will be enlarged as needed to avoid clipping.

28.12.7 Translate...

Translates (moves) the image in the x and y directions by a specified number of pixels. With stacks, you can translate either the current image or all the images in the stack. Two resampling methods are possible: Bilinear and Bicubic interpolation (cf. Image ⊲ Size...).

See also: Align_Slice and Align_RGB_planes plugins

28.12.8 Bin...

Reduces the size of an image by binning groups of pixels of user-specified size \((X, Y, Z\) shrink factor) \([27]\). The resulting pixel can be calculated as Average, Median, Maximum, or Minimum. Undo support is restricted to 2D images (non stacks).

Z binning produces equivalent results to Image ⊲ Stacks ⊲ Tools ⊲ Grouped Z Project... However, there are two main differences between the two commands: While Bin... replaces the original image, Grouped Z Project... creates a new substack, and while Z shrink factor takes any value, Group size must divide evenly into the stack size.

See also: Image ⊲ Adjust ⊲ Size..., Image ⊲ Adjust ⊲ Scale... [E], Image ⊲ Stacks ⊲ Tools ⊲ Grouped Z Project...

28.12.9 Image to Results

Prints the active area selection to the Results Table, clearing previous results. The entire image is processed when no area ROI exists. XY coordinates are detailed in column and row headers. Calibrated and floating-point images are listed with the precision specified by Decimal places in Analyze ⊲ Set Measurements...

For RGB images, each pixel is converted to grayscale using the formula \(gray = (red + green + blue) / 3\) or the formula \(gray = 0.299 \times red + 0.587 \times green + 0.114 \times blue\) if Weighted RGB to Grayscale Conversion is checked in Edit ⊲ Options ⊲ Conversions...

See also: Text Images, Pixel Inspector, Results to Image, File ⊲ Save As ⊲ Text Image..., Import ⊲ Text Image..., Stack_to_Results.js, a script that displays the contents of a stack using one image per column

28.12.10 Results to Image

The reverse of Image to Results, converting the tabular data in the Results Table into a 32-bit image named Results Table. Column and row headers are ignored.

See also: Text Images, Pixel Inspector, File ⊲ Save As ⊲ Text Image..., Import ⊲ Text Image...

28.13 Zoom

This submenu contains commands that control how the current image is displayed. The \(\uparrow\) and \(\downarrow\) or \(\rightarrow\) and \(\leftarrow\) keys are the preferred way to use the In [+] and Out [−] commands. When a selection exists, zooming with the Arrow Keys requires holding down either Shift or Ctrl.

28.13.1 In [+] Zooms in to next higher magnification level and, if possible, enlarges the window. As explained in Magnifying Glass, there are 21 possible levels (shown in the title bar): 3.1, 4.2, 6.3, 8.3, 12.5, 16.7, 25, 33.3, 50, 75, 100, 150, 200, 300, 400, 600, 800, 1200, 1600, 2400 and 3200%.

28.13.2 Out [−] Zooms out to next lower magnification level and, if needed, shrinks the window.

28.13.3 Original Scale [4] Displays the image at the magnification used when the image was first opened. As a shortcut, double click on the Magnifying Glass tool.
Images are magnified using \(+\) and \(-\), or \(\uparrow\) and \(\downarrow\) if no selection exists. Magnification occurs around the cursor, or to the center of the image when the cursor lays outside the image canvas. The Zoom indicator in the upper left corner of magnified images shows what portion of the image is currently displayed. At high magnification levels the pixel grid becomes visible by default unless Interpolate zoomed images is checked in Edit > Options > Appearance. To scroll a magnified image, hold down the space bar (Scrolling Tool shortcut) while dragging the cursor.

By default, Overlays and the active selection are displayed with a 1–pixel wide contour regardless of the image magnification (\(0 Width\)). To thicken ROI edges at higher zoom levels, set Stroke width to a non-zero value in Edit > Selection > Properties... [y], Image > Overlay > Overlay Options... or ROI Manager's Properties... [y].

Selections are immediately added to the current overlay when pressing \([\text{B}]\). Pressing \([\text{Alt}\,\text{B}]) will display a dialog box in which is possible to remove previously added Overlays. Outputs from Grid_Overlay, MakeOverlay and ROI Color Coder that exemplify the usage of most Image>Overlay>submenus commands. The Overlay Brush tool can also be used to create freehand annotations.

28.14.1 Add Selection... [b]

Selections are immediately added to the current overlay when pressing \([\text{B}]\). Pressing \([\text{Alt}\,\text{B}]) will display a dialog box in which Stroke color and Width and Fill color can be set.

Except for text selections, and as explained in Edit > Selection > Properties... [y], Stroke (contour) color and Width are ignored if a Fill color is specified. Colors are specified using the name of one of the default selection colors (black, blue, cyan, green, magenta, orange, red, white and yellow) or using hex notation (see XIX Hexadecimal Color Values).

Previously added Overlays are removed if New overlay is checked. Also, if no selection exists and the command is run, a warning message is displayed in which is possible to remove the existing overlay, by running Remove Overlay.

Note that measured selections (Analyze=>Measure... [m]) can be added automatically to the image overlay by selecting the Add to overlay checkbox in Analyze=>Set Measurements... [m].

28.14.2 Add Image...

Blends two open images by adding an image to the overlay of frontmost image. The image to be blended can be of any type (see Image Types and Formats) but cannot be larger than the host image. A blending alpha value can be specified in the Opacity (0–100%) field. The initial X,Y location is based on the existing rectangular selection, if any.
By default the created overlay image cannot be moved around the canvas, i.e., is not a image selection (ImageROI), but are stored in the TIFF header and can be saved and restored when saving images in TIFF format. On the other hand, image selections can be created using Edit>Selection>Image to Selection... or by running Image>Overlay> To ROI Manager after adding the image to the overlay.

**XIX Hexadecimal Color Values**

Hexadecimal (hex) notation is frequently used in computing to summarize binary code in a human-friendly manner. Here are some decimal/hexadecimal equivalents:

<table>
<thead>
<tr>
<th>Dec</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hex</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td>00</td>
<td>01</td>
<td>02</td>
<td>03</td>
<td>04</td>
<td>05</td>
<td>06</td>
<td>07</td>
<td>08</td>
<td>09</td>
<td>0A</td>
<td>0B</td>
<td>0C</td>
<td>0D</td>
<td>0E</td>
<td>0F</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>1A</td>
<td>1B</td>
<td>1C</td>
<td>1D</td>
<td>1E</td>
<td>1F</td>
</tr>
</tbody>
</table>

Decimal RGB color values that typically range from 0 to 255 are succinctly represented by two hexadecimal digits ranging from 00 through FF. A hex color value is a 6-digit, three-byte hexadecimal number (hex triplet) in the form #RRGGBB, in which RR specifies the red, GG the green and BB the blue value.

Opacity of hex triplets can be modified by an optional transparency (alpha) channel, giving rise to a 8-digit, four-byte hex number in the form #AARRGGBB with AA specifying the alpha blending value. In ImageJ, the alpha channel codes for opacity and ranges from 00 (fully transparent) to FF (solid color). Alpha values can be omitted for fully opaque colors. The table below provides some hexdecimal color values in 10% transparency increments, and can be used to create semi-transparent Overlays.

<table>
<thead>
<tr>
<th>Color</th>
<th>RGB triplet</th>
<th>Opacity value</th>
<th>Hex triplet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blue</td>
<td>0, 0, 255</td>
<td>100%</td>
<td>#0000FF</td>
</tr>
<tr>
<td>Cyan</td>
<td>0, 255, 255</td>
<td>90%</td>
<td>#00FFFF</td>
</tr>
<tr>
<td>Green</td>
<td>255, 0, 0</td>
<td>80%</td>
<td>#FF0000</td>
</tr>
<tr>
<td>Magenta</td>
<td>255, 255, 0</td>
<td>70%</td>
<td>#FFCCFF</td>
</tr>
<tr>
<td>Red</td>
<td>255, 0, 0</td>
<td>60%</td>
<td>#FF00FF</td>
</tr>
<tr>
<td>Orange</td>
<td>255, 150, 0</td>
<td>50%</td>
<td>#FF9900</td>
</tr>
<tr>
<td>Yellow</td>
<td>255, 255, 0</td>
<td>40%</td>
<td>#FFFF00</td>
</tr>
<tr>
<td>White</td>
<td>255, 255, 255</td>
<td>30%</td>
<td>#FFFFFF</td>
</tr>
<tr>
<td>Gray</td>
<td>127, 127, 127</td>
<td>20%</td>
<td>#7F7F7F</td>
</tr>
<tr>
<td>Black</td>
<td>0, 0, 0</td>
<td>10%</td>
<td>#000000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0%</td>
<td>#00000000</td>
</tr>
</tbody>
</table>

Alpha blending values can be added to the beginning of hex triplets to modulate color transparency: e.g., #7F000000 defines red at 50% opacity. In ImageJ the hash (#) prefix is optional.

Color Choosers in the Color Picker window display hex values of RGB colors. The built-in macro function toText() returns hexadecimal representations of decimal numbers and can also be used to convert RGB color values (see RGBtoHEX macro). Several other macros (e.g., MakeOverlay) exemplify how to annotate images using hex colors.

**28.14.3 Hide Overlay**

Causes ImageJ to stop displaying the overlay displayed by Show Overlay.

**28.14.4 Show Overlay**

Displays an overlay that was hidden by Hide Overlay.

**28.14.5 From ROI Manager**

Creates an overlay from the selections on the ROI Manager list (see Analyze>Tools> ROI Manager...). Note that previously added Overlays will be removed.

**28.14.6 To ROI Manager**

Copies the selections and images in the current overlay to the ROI Manager, where they can be edited (moved, resized or re-colored) (see ROI Manager...). Note that previous items in the ROI Manager list will be deleted.

**28.14.7 Remove Overlay**

Permanently clears the overlay so that it cannot be restored using Show Overlay.

**28.14.8 Flatten [F]**

Creates a new RGB image that has the overlay rendered as pixel data. The RGB image is the same size as the active image, unlike Plugins>Utilities> Capture Image, which creates a WYSIWYG (What You See Is What You Get) image that is the same size as its window. Stacks must first be converted to RGB (Image> Type> submenu, see also Image Types and Formats) when flattening all slices in the stack.

**28.14.9 Labels...**

This prompt defines if and how Overlays should be labelled.

It sets the behavior of Image> Overlay> Add Selection... [b], Analyze> Measure... [m] (after activating the Add to overlay option in Analyze> Set Measurements...), Analyze> Analyze Particles... and the ROI Manager Show All display.

**Color** The color of the label as one of the default selection colors.

**Font size** Specifies the font size of the label (12–72 pt).
Show labels If overlays should be decorated with a text label. This option is inactive by default.

Use names as labels If checked, ROI names are used instead of the default numeric labels. If unchecked, the size (selection count) of the current overlay is used. Selections can be renamed using the Edit>Selection>Properties... [y] command, or, when using the ROI Manager, either Rename... or Properties...

Draw backgrounds If checked, text will be displayed on complementary colored background. This option produces similar labels to those produced by the ROI Manager Show All option when Labels is activated.

Bold If checked, labels (typeset in sans-serif font) are displayed in boldface.

See also: Overlay Options..., Add Selection... [b]

28.14.10 Overlay Options...

Use this command to define the default overlay Stroke color, Stroke width and Fill color. As mentioned in Edit>Selection>Properties... [y] and Image>Overlay>Add Selection... [b], Stroke (contour) color and Width are ignored if a Fill color is specified. Set Stroke width to 0 to have selections drawn using a width of one pixel regardless of the image magnification (see XVIII Working with Zoomed Canvases).

As usual, colors are specified using the name of one of the ImageJ default colors (black, blue, cyan, green, magenta, orange, red, white and yellow) or using Hexadecimal Color Values.

With Stacks and Hyperstacks, selecting the Set stack positions checkbox will make Overlays visible only when browsing their respective slice or frame. If unchecked, overlays will be displayed throughout the stack (see also ROI Manager..., More>Options...).

See also: Labels..., Remove Slice Info (Analyze>Tools>ROI Manager...), Settings and Preferences

28.15 Lookup Tables

This submenu contains a selection of color lookup tables that can be applied to grayscale images to produce Pseudocolor Images. In addition, it lists all the lookup tables installed in the ImageJ/luts/ directory. More than 100 additional lookup tables are available from the ImageJ website as individual files or, in bulk, as a ZIP archive.

As explained earlier, it is not possible to organize LUTs into subfolders. However, the most frequently used lookup tables can be renamed with a numeric prefix (e.g., 01-glasbey.lut, 02-Termal.lut, etc.) so that they are listed earlier in the menu. This submenu can also be accessed from the Toolbar by loading the LUT Menu.

When loading a lookup table is loaded and no image is open, a 256×32 ramp image is created to display the color table.

See also: Channels... widget, Show All LUTs, a macro that creates a graphical palette of all the installed lookup tables.

28.15.1 Invert LUT

Inverts the current lookup table. For 8-bit images, the value (v) of each entry in the table is replaced by 255 − v. With inverted LUTs, pixels with a value of zero are white and pixels with a value 255 are black. Unlike the Edit>Invert [i] command, pixels values are not altered, only the way the image is displayed on the screen.

See also: Image>Color>Show LUT/Edit LUT...

28.15.2 Apply LUT

Applies the current lookup table function to each pixel in the image or selection and restores the default identity function. This modifies the gray values so that when the image is viewed using the default grayscale lookup table it will look the same as it did before. This command is equivalent to clicking on Apply in Image>Adjust>Brightness/Contrast... [C]. For thresholded images, it is equivalent to clicking on Apply in Image>Adjust>Threshold... [T].

See also: Image>Color>Show LUT/Edit LUT...
This menu lists all commands related to image processing, including point operations, filters, and arithmetic operations between multiple images [11]. The File→Open Samples→Blobs (25K) [B] image will be used in most of the illustrations of this section.

29.1 Smooth [S]
Blurs the active image or selection. This filter replaces each pixel with the average of its 3×3 neighborhood.

29.2 Sharpen
Increases contrast and accentuates detail in the image or selection, but may also accentuate noise. This filter uses the following weighting factors to replace each pixel with a weighted average of the 3×3 neighborhood:

\[
\begin{array}{ccc}
-1 & -1 & -1 \\
-1 & 12 & -1 \\
-1 & -1 & -1 \\
\end{array}
\]

29.3 Find Edges
Uses a Sobel edge detector to highlight sharp changes in intensity in the active image or selection. Two 3×3 convolution kernels (shown below) are used to generate vertical and horizontal derivatives. The final image is produced by combining the two derivatives using the square root of the sum of the squares.

\[
\begin{array}{ccc}
1 & 2 & 1 \\
0 & 0 & 0 \\
-1 & -2 & -1 \\
\end{array}
\]

\[
\begin{array}{ccc}
1 & 0 & -1 \\
2 & 0 & -2 \\
1 & 0 & -1 \\
\end{array}
\]

29.4 Find Maxima…
Determines the local maxima in an image and creates a binary (mask-like) image of the same size with the maxima, or one segmented particle per maximum, marked [29]. Analysis is performed on the existing rectangular selection or on the entire image if no selection is present.

For RGB images, maxima of luminance are selected, with the luminance defined as weighted or unweighted average of the colors depending on how Weighted RGB to Grayscale Conversion is set in Edit→Options→Conversions….

Output Type
Can be (see Process→Find Maxima… outputs):

- **Single Points** Results in one single point per maximum.
- **Maxima Within Tolerance** All points within the Noise Tolerance for each maximum.
- **Segmented Particles** Assumes that each maximum belongs to a particle and segments the image by a watershed algorithm applied to the values of the image (in contrast to Process→Binary→Watershed, which uses the Euclidian distance map – EDM). See Process→Binary→Voronoi for EDM-based segmentation of binary images.
- **Point Selection** Displays a multi-point selection with a point at each maximum.
- **List** Displays the XY coordinates of each maximum in the Results window.
- **Count** Displays the number of maxima in the Results window.

Exclude Edge Maxima Excludes maxima if the area within the noise tolerance surrounding a maximum touches the edge of the image (edge of the selection does not matter).

Above Lower Threshold (Thresholded images only) Finds maxima above the lower threshold only. The upper threshold of the image is ignored. If Segmented Particles is selected as Output Type, the area below the lower threshold is considered a background. This option cannot be used when finding minima (image with light background and inverted LUT).

Light Background To be checked if the image background is brighter than the objects to be found, as it is in the Cell Colony image in the illustration below.

Enhance Contrast...

Enhances image contrast by using either histogram stretching or histogram equalization. Both methods are described in detail in the Hypermedia Image Processing Reference – Contrast Stretching and Histogram Equalization.

This command does not alter pixel values as long as Normalize, Equalize Histogram or Normalize All n Slices (in the case of stacks) are not checked.

**Saturated Pixels** Determines the number of pixels in the image that are allowed to become saturated. Increasing this value will increase contrast. This value should be greater than zero to prevent a few outlying pixel from causing the histogram stretch to not work as intended.

**Normalize**: If checked, ImageJ will recalculate the pixel values of the image so the range is equal to the maximum range for the data type, or 0–1.0 for float images. The contrast stretch performed on the image is similar to the ‘Auto’ option in the Brightness/Contrast... [C] window, except that with stacks, each slice in the stack is adjusted independently, according to the optimal for that slice alone (if Use Stack Histogram is unchecked). The maximum range is 0–255 for 8-bit images and 0–65535 for 16-bit images. With stacks another checkbox, Normalize All n Slices, is displayed. If checked, normalization will be applied to all slices in the stack. Note that normalization of RGB images is not supported, and thus this option will not be available on RGB stacks.

**Equalize Histogram**: If checked, ImageJ will enhance the image using histogram equalization [30]. Create a selection and the equalization will be based on the histogram of that selection.

Uses a modified algorithm that takes the square root of the histogram values. Hold Alt to use the standard histogram equalization algorithm. The Saturated Pixels and Normalize parameters are ignored when Equalize Histogram is checked.

**Use Stack Histogram** If checked, ImageJ will use the overall stack histogram instead of individual slice histograms, that allow optimal adjustments for each slice alone. This option may be specially relevant when performing enhancements based on a ROI.


See also: *Brightness/Contrast... [C]*, XIV Applying Auto Brightness/Contrast to Entire Stacks

29.6 Noise

Use the commands in this submenu to add noise to images or remove it.

**Process**: Add Noise, Salt and Pepper, Despeckle, Add Noise and Remove Outliers...

See also: RandomJ (Binomial, Exponential, Gamma, Gaussian, Poisson and Uniform) a Java package for image randomization by Erik Meijering
29.6.4 Despeckle

This is a median filter. It replaces each pixel with the median value in its $3 \times 3$ neighborhood. This is a time consuming operation because, for each pixel in the selection, the nine pixels in the $3 \times 3$ neighborhood must be sorted and the center pixel replaced with the median value (the fifth). Median filters are good at removing salt and pepper noise.

29.6.5 Remove Outliers...

Replaces a pixel by the median of the pixels in the surrounding if it deviates from the median by more than a certain value (the threshold). Useful for correcting, e.g., hot pixels or dead pixels of a CCD camera.

*Radius* Determines the area (uncalibrated, i.e., in pixels) used for calculating the median. Run *Process > Filters > Show Circular Masks* ... to see how radius translates into an area.

*Threshold* Determines by how much the pixel must deviate from the median to get replaced.

*Which Outliers* Determines whether pixels brighter or darker than the surrounding (the median) should be replaced.

See also: Despeckle

29.6.6 Remove NaNs...

This filter replaces NaN (Not-a-Number) pixels in 32-bit (float) images by the median of the neighbors inside the circular kernel area defined by *Radius* [31]. It does not remove patches of NaNs larger than the kernel size, however.

Note that some ImageJ filters, such as *Process > Filters > Gaussian Blur*..., *Mean*..., and *Variance*... destroy the surrounding of NaN pixels by setting it also to NaN. Other filters may produce invalid results in the position of NaN pixels.

See also: NaNs.txt, a macro that demonstrates how to create, count and remove NaNs

29.7 Shadows

Commands in this submenu produce a shadow effect, with light appearing to come from a direction corresponding to the command name (*East*, *North*, *Northeast*, *Northwest*, *South*, *Southeast*, *Southwest* and *West*). The commands use Convolve $3 \times 3$. ImageJ’s $3 \times 3$ convolution function. The *Shadows Demo* command uses all eight kernels to demonstrate the speed of Convolve $3 \times 3$. The illustration below shows four of the *Shadows* convolution kernels.

29.8 Binary

This submenu contains commands that create or process binary (black and white) images. They assume that objects are black and background is white unless *Black Background* is checked in the *Process > Binary > Options*... dialog box (see XXII Interpreting Binary Images). The commands use Convolve $3 \times 3$. The illustration below shows four of the *Shadows* convolution kernels.

**Summary of morphological operators (Process > Binary > submenu).**

<table>
<thead>
<tr>
<th>ImageJ</th>
<th>Make Binary</th>
<th>Erode</th>
<th>Dilate</th>
<th>Open</th>
<th>Close</th>
<th>Outline</th>
<th>Skeletonize</th>
</tr>
</thead>
</table>

Adjust Threshold... [T] | Minimum... (grayscale) | Maximum... (grayscale) | Erode then Dilate | Dilate then Erode | 1 pixel wide outline | 1 pixel wide skeleton

29.8.1 Make Binary

Converts an image to black and white. If a threshold has been set using the *Image > Adjust > Threshold*... [T] tool, the depicted dialog is displayed. The value of the *Black foreground, white background* checkbox reflects and sets the global *Black Background* value of *Process > Binary > Options*...

If a threshold has not been set, Make Binary will analyze the histogram of the current selection, or of the entire image if no selection is present, and set an automatic threshold level to create the binary image (‘Auto-thresholding’ is displayed in the Status bar, cf. *Threshold*... [T]).

With stacks the Convert to Mask dialog box is displayed. Note that for non-thresholded images and stacks Make Binary and Convert to Mask behave similarly.

See also: *Edit > Selection > Convert to Mask*, XX Creating Binary Masks, XXII Interpreting Binary Images
**29.8.2 Convert to Mask**

Converts an image to black and white. The mask will have an inverting LUT (white is 0 and black is 255) unless Black Background is checked in the Process\>Binary\>Options... dialog box. If a threshold has not been set, automatic threshold levels will be calculated (cf. Make Binary). Note that for non-thresholded images and stacks Make Binary and Convert to Mask behave similarly.

With stacks, the depicted dialog is displayed.

*Calculate Threshold for Each Image* If checked, threshold levels will be calculated for each individual slice, otherwise the calculated threshold of the currently displayed slice will be used for all slices.

*Black Background* Defines whether the background is black and the foreground is white.

Note that the value of this checkbox reflects and sets the global Black Background value of Process\>Binary\>Options...

[See also: Make Binary, Edit\>Selection\>Create Mask, XX Creating Binary Masks, XXII Interpreting Binary Images]

---

**29.8.3 Erode**

Removes pixels from the edges of objects in a binary image. Use Filters\>Minimum... to perform grayscale erosion on non-thresholded images.

[See also: Binary\>Options..., XXII Interpreting Binary Images]

---

**29.8.4 Dilate**

Adds pixels to the edges of objects in a binary image. Use Filters\>Maximum... to perform grayscale dilation on non-thresholded images.

[See also: Binary\>Options..., XXII Interpreting Binary Images]

---

**29.8.5 Open**

Performs an erosion operation, followed by dilation. This smooths objects and removes isolated pixels.

[See also: Binary\>Options..., XXII Interpreting Binary Images]

---

**29.8.6 Close**

Performs a dilation operation, followed by erosion. This smooths objects and fills in small holes. The command has a trailing hyphen to differentiate it from File\>Close [w].

[See also: Binary\>Options..., XXII Interpreting Binary Images]

---

**29.8.7 Outline**

Generates a one pixel wide outline of foreground objects in a binary image. The line is drawn inside the object, i.e., on previous foreground pixels.

[See also: Binary\>Options..., XXII Interpreting Binary Images]

---

**29.8.8 Fill Holes**

This command fills holes (4-connected background elements) in objects by filling the background.

[See also: Binary\>Options..., XXII Interpreting Binary Images]

---

**29.8.9 Skeletonize**

Repeatably remove pixels from the edges of objects in a binary image until they are reduced to single-pixel-wide shapes (topological skeletons). As explained in XXI Skeletonize vs Skeletonize 3D, there are several skeletonization algorithms. ImageJ implements a thinning algorithm from Zhang and Suen. A fast parallel algorithm for thinning digital patterns. CACM 27(3):236–239.

[See also: Binary\>Options..., XXII Interpreting Binary Images]
1984, in which a lookup table indexes all the 256 possible 3×3 neighborhood configurations for each foreground pixel. The algorithm calculates the index number for each object pixel, and uses the lookup table to decide if the pixel is eliminable. This process is repeated until no pixel can be eliminated.

When debugging is enabled in Edit>Options>Misc..., Skeletonize creates an animation documenting the iterations of the thinning algorithm.

SEE ALSO: AnalyzeSkeleton plugin, BinaryProcessor source code

### XXI Skeletonize vs Skeletonize 3D

Skeletonize3D is a ImageJ plugin written by Ignacio Arganda-Carreras [35] that offers several advantages over Process>Binary>Skeletonize, the legacy skeletonization algorithm of ImageJ:

- Skeletonize works only with binary 2D images. Skeletonize3D works with 8-bit 2D images and stacks, expecting the image to be binary. If not, Skeletonize3D considers all pixel values above 0 to be white (255).
- While Skeletonize relies on Black background value in Binary>Options... (see XXII Interpreting Binary Images), the output of Skeletonize3D always has a value of 255 at the skeleton and 0 at background pixels, independently of the Black background option.

In Fiji, Skeletonize 3D is already pre-installed as Plugins>Skeleton>Skeletonize (2D/3D). In ImageJ, it can be downloaded and installed from the Skeletonize3D homepage.

### 29.8.10 Distance Map

Generates a Euclidian distance map (EDM) from a binary image [69]. Each foreground pixel in the binary image is replaced with a gray value equal to that pixel's distance from the nearest background pixel (for background pixels the EDM is 0). The Ultimate Points, Watershed and Voronoi operations are based on the EDM algorithm.

The output type (Overwrite, 8-bit, 16-bit or 32-bit) of this command can be set in the Binary>Options... dialog box. Note that when selecting 'Overwrite' or '8-bit output', distances larger than 255 are labelled as 255.

### 29.8.11 Ultimate Points

Generates the ultimate eroded points (UEPs) of the Euclidian distance map (EDM, see Distance Map) from a binary image. Ultimate Eroded Points are maxima of the EDM. In the output, the points are assigned the EDM value, which is equal to the radius of the largest circle that fits into the binary particle, with the UEP as the center. The output type (Overwrite, 8-bit, 16-bit or 32-bit) of this command can be set in the Binary>Options... dialog box.

### 29.8.12 Watershed

Watershed segmentation is a way of automatically separating or cutting apart particles that touch. It first calculates the Euclidian distance map (EDM) and finds the ultimate eroded points (UEPs). It then dilates each of the UEPs (the peaks or local maxima of the EDM) as far as possible – either until the edge of the particle is reached, or the edge touches a region of another (growing) UEP. Watershed segmentation works best for smooth convex objects that don't overlap too much.

Enable debugging in Edit>Options>Misc... and the Watershed command will create an animation that shows how the watershed algorithm works (cf. online example).

SEE ALSO: Find Maxima... (Segmented Particles output) for watershed segmentation of grayscale images.

### 29.8.13 Voronoi

Splits the image by lines of points having equal distance to the borders of the two nearest particles. Thus, the Voronoi cell of each particle includes all points that are nearer to this particle than any other particle. When particles are single points, this process is a Voronoi tessellation (also known as Dirichlet tessellation).

The output type (Overwrite, 8-bit, 16-bit or 32-bit) of this command can be set in the Process>Binary>Options... dialog box. In the output, the value inside the Voronoi cells is zero; the pixel values of the dividing lines between the cells are equal to the distance between the two nearest particles. This is similar to a medial axis transform of the background, but there are no lines in inner holes of particles.

SEE ALSO: Find Maxima... (Segmented Particles output), Delaunay _ Voronoi plugin

### 29.8.14 Options...)

Specifies several settings used by Binary> commands.

- **Iterations** Specifies the number of times erosion, dilation, opening, and closing are performed. Iterations can be aborted by pressing Esc.
- **Count** Specifies the number of adjacent background pixels necessary before a pixel is removed from the edge of an object during erosion and the number of adjacent foreground pixels necessary before a pixel is added to the edge of an object during dilation.
Black background If checked, binary images will be created without using an inverted LUT (cf. XX Creating Binary Masks) and commands in the Process▷Binary▷submenu will assume that images contain white objects on a black background (see XXII Interpreting Binary Images). Macros can set this option using setOption("BlackBackground", true); (see XXII Interpreting Binary Images and Settings and Preferences).

Pad edges when eroding If checked, Binary▷Erode does not erode from the edges of the image. This setting also affects Binary▷Close▷, which erodes from the edges unless this checkbox is selected.

EDM output Determines the output type for the Binary▷Distance Map, Ultimate Points and Voronoi commands. Set it to ‘Overwrite’ for 8-bit output that overwrites the input image; ‘8-bit’, ‘16-bit’ or ‘32-bit’ for separate output images. 32-bit output has floating point (subpixel) distance resolution.

Do This drop-down menu allows one to test the chosen settings by previewing each binary operation (Erode, Dilate, Open, Close, Outline, Fill Holes, Skeletonize) on the active image. This option is only available when the active image is binary.


29.9 Math▷

The commands in this submenu add (subtract, multiply, etc.) a constant to each pixel in the active image or selection. A ‘Preview’ option is available for most operations.

With stacks, the dialog depicted on the left is displayed. Choose ‘Yes’ to process entire stack or ‘No’ to process only the active slice. The dialog is not displayed if Hide "Process Stack?" dialog is checked in Edit▷Options▷Misc….

See also: Memory & Threads…

29.9.1 Add…

Adds a constant to the image or selection. With 8-bit images, results greater than 255 are set to 255. With 16-bit signed images, results greater than 65,535 are set to 65,535.

29.9.2 Subtract…

Subtracts a constant from the image or selection. With 8-bit and 16-bit images, results less than 0 are set to 0.

29.9.3 Multiply…

Multiplies the image or selection by the specified real constant. With 8-bit images, results greater than 255 are set to 255. With 16-bit signed images, results greater than 65,535 are set to 65,535.

29.9.4 Divide…

Divides the image or selection by the specified real constant. Except for 32-bit (float) images, attempts to divide by zero are ignored. With 32-bit images, dividing by zero results in Infinity, -Infinity or NaN (0/0) pixels when the source pixels are positive, negative or zero. The divide-by-zero value can be redefined using Edit▷Options▷Misc….

29.9.5 AND…

Does a bitwise AND of the image and the specified binary constant.

29.9.6 OR…

Does a bitwise OR of the image and the specified binary constant.

29.9.7 XOR…

Does a bitwise XOR of the image and the specified binary constant.
29.9.8 Min...
Pixels in the image with a value less than the specified constant are replaced by the constant.

29.9.9 Max...
Pixels in the image with a value greater than the specified constant are replaced by the constant.

29.9.10 Gamma...
Applies the function $f(p) = (p/255)^\gamma \times 255$ to each pixel $p$ in the image or selection, where $0.1 \leq \gamma \leq 5.0$. For RGB images, this function is applied to all three color channels. For 16-bit images, the image min and max are used for scaling instead of 255.

See also: GammaCorrectionTool macro

29.9.11 Set...
Fills the image or selection with the specified value.

29.9.12 Log
For 8-bit images, applies the function $f(p) = \ln(p) \times 255 / \ln(255)$ to each pixel $p$ in the image or selection. For RGB images, this function is applied to all three color channels. For 16-bit images, the image min and max are used for scaling instead of 255. For float images, no scaling is done. To calculate $\log_{10}$ of the image, multiply the result of this operation by $0.4343$ ($1 / \ln(10)$).

29.9.13 Exp
Performs an exponential transform on the active image or selection.

29.9.14 Square
Performs a square transform on the active image or selection.

29.9.15 Square Root
Performs a square root transform on the active image or selection.

29.9.16 Reciprocal
Generates the reciprocal (multiplicative inverse) of the active image or selection, transforming each pixel $p$ into $1/p$. Requires 32-bit float images (see Image Types and Formats).

29.9.17 NaN Background
Sets non-thresholded pixels in 32-bit float images to the NaN (Not a Number) value. For float images, the Apply option in Image▶Adjust▶Threshold... [T] runs this command.

Pixels with a value of Float.NaN (0f/0f), Float.POSITIVE_INFINITY (1f/0f) or Float.NEGATIVE_INFINITY (-1f/0f) are ignored when making measurements on 32-bit float images.

29.9.18 Abs
Generates the absolute value of the active image or selection. Works only with 32-bit float or signed 16-bit image images.

29.9.19 Macro...
This command performs image arithmetic using an expression specified by the user [32]. It can be used to create fully-synthetic images or to perform precise pixel manipulations on existing images. The MathMacroDemo macro demonstrates the usage of this command.

See also: Expression plugin

29.10 FFT
The commands in this submenu support frequency domain display, editing and processing. They are based on an implementation of the 2D Fast Hartley Transform (FHT) contributed by Arlo Reeves, the author of the ImageFFT, spinoff of NIH Image[1]. 3D FHT can be performed using Bob Dougherty’s 3D Fast Hartley Transform plugin.

The frequency domain image is stored as 32-bit float FHT attached to the 8-bit image that displays the power spectrum. Commands in this submenu, such as Inverse FFT, operate on the 32-bit FHT, not on the 8-bit power spectrum. All other ImageJ commands only ‘see’ the power spectrum.

Two FFT dedicated tutorials are available on the ImageJ website: FFT Measurements and FFT Filtering.

29.10.1 FFT
Computes the Fourier transform and displays the power spectrum. Polar coordinates of measured point selections are recorded by Analyze▶Measure... [m].

If the mouse is over an active frequency domain (FFT) window, its location is displayed in polar coordinates. The angle is expressed in degrees, while the radius is expressed in pixels per cycle (p/c). The radius is expressed in [units] per cycle (e.g. mm/c) if the spatial scale of the image was defined using Image▶Properties... [P] or Analyze▶Set Scale...[1]

[1] Although outdated, the ImageFFT documentation summarizes important frequency domain methodologies.
29.10.2 Inverse FFT

Computes the inverse Fourier transform. You can filter or mask spots on the transformed (frequency domain) image and do an inverse transform to produce an image which only contains the frequencies selected or which suppresses the frequencies selected. Use ImageJ’s selection tools and fill/clear commands to draw black or white areas that mask portions of the transformed image. Black areas (pixel value = 0) cause the corresponding frequencies to be filtered (removed) and white areas (pixel value = 255) cause the corresponding frequencies to be passed. It is not, however, possible to both filter and pass during the same inverse transform.

Note that areas to be filtered in the frequency domain image must be zero filled and areas to be passed must be filled with 255. You can verify that this is the case by moving the cursor over a filled area and observing that the values displayed in the status bar are either 0 or 255. Thus, you should always confirm that masked areas are not some other gray value, by using the black & white reset option in the Color Picker window widgets when defining foreground (Edit ⊿ Fill [f]) and background (Edit ⊿ Clear) colors.

FFT: Example of low frequencies filtering.

With off-center selections, the same spatial frequency appears twice in the power spectrum, at points opposite from the center. It is sufficient to fill / clear only one of these.

29.10.3 Redisplay Power Spectrum

Recomputes the power spectrum from the frequency domain image (32-bit FHT). This command allows you to start over after mis-editing the 8-bit power spectrum image.

29.10.4 FFT Options...

Displays the FFT Options dialog box. The first group of checkboxes specifies which image(s) are created by the FFT command:

Display FFT Window The standard output. It consists of an 8-bit image of the power spectrum and the actual data, which remain invisible for the user. The power spectrum image is displayed with logarithmic scaling, enhancing the visibility of components that are weakly visible. The actual data are used for the Inverse FFT command.

Display Raw Power Spectrum The power spectrum without logarithmic scaling.

Display Fast Hartley Transform The internal format used by the command, which is based on a Hartley transform rather than Fourier transform.

Display Complex Fourier Transform A stack with two slices for the real and imaginary parts of the FFT.

Do Forward Transform If checked, the current image is transformed immediately when closing the FFT Options dialog.

29.10.5 Bandpass Filter...

Removes high spatial frequencies (blurring the image) and low spatial frequencies (similar to subtracting a blurred image). It can also suppress horizontal or vertical stripes that were created by scanning an image line by line [33]. The Bandpass Filter uses a special algorithm to reduce edge artifacts (before the Fourier transform, the image is extended in size by attaching mirrored copies of image parts outside the original image, thus no jumps occur at the edges).

Filter Large Structures Down to Smooth variations of the image with typical sizes of bright or dark patches larger than this value are suppressed (background).

Filter Large Structures Up to Determines the amount of smoothing. Objects in the image smaller than this size are strongly attenuated. Note that these values are both half the spatial frequencies of the actual cutoff. The cutoff is very soft, so the bandpass will noticeably attenuate even spatial frequencies in the center of the bandpass unless the difference of the two values is large (say, more than a factor of 5 or so).

Suppress Stripes Select whether to eliminate Horizontal or Vertical stripes. Removal of horizontal stripes is similar to subtracting an image that is only blurred in the horizontal direction from the original.

Tolerance of Direction This is for Suppress Stripes; higher values remove shorter stripes and/or stripes that are running under an angle with respect to the horizontal (vertical) direction.
**Autoscale After Filtering** If checked, puts the lowest intensity to 0 and the highest intensity to 255, preserving all intensities.

**Saturate Image when Autoscaling** If checked, allows some intensities to go into saturation, and produces a better visual contrast. Saturate Image when Autoscaling only has an effect when Autoscale After Filtering is enabled.

**Display Filter** If checked, shows the filter generated. Note that this disables Undo of the filter operation on the original image.

### 29.10.6 Custom Filter...

This command does Fourier space filtering of the active image using a user-supplied spatial domain (non-FFT) image as the filter.

This image will be converted to 8-bit. For pixels that have a value of 0, the corresponding spatial frequencies will be blocked. Pixel with values of 255 should be used for passing the respective spatial frequencies without attenuation. Note that the filter should be symmetric with respect to inversion of the center: Points that are opposite of the center point (defined as $x = \text{width}/2$, $y = \text{height}/2$) should have the same value. Otherwise, artifacts can occur. For some examples, see the FFTCustomFilterDemo and FFTRemoveStreaks macros.

### 29.10.7 FD Math...

This command correlates, convolves or deconvolves two images.

It does this by converting Image1 and Image2 to the frequency domain, performing conjugate multiplication or division, then converting the result back to the space domain. These three operations in the frequency domain are equivalent to convolution, correlation and deconvolution in the space domain.

Refer to the DeconvolutionDemo and MotionBlurRemoval macros for examples.

### 29.10.8 Swap Quadrants

This command transforms between the ‘user friendly’ display of Fourier transforms with the lowest frequencies at the center and the ‘native’ form with the lowest frequencies at the four corners.

Swap Quadrants swaps quadrants I with III and II with IV (counterclockwise starting from ‘Northeast’) so that points near the center are moved towards the edge and vice versa. Another way to see this command is to imagine that the image is periodically repeated and the origin is shifted by width/2 in $x$ and by height/2 in $y$ direction.

For Fourier transforms, Swap Quadrants affects only the image displayed, not the actual FHT data. Therefore, editing an image with swapped quadrants for filtering or masking may lead to undesired results.

### 29.11 Filters...

This submenu contains miscellaneous filters [30] (including those installed by the Utilities⇒Install Plugin… command).

More information on image filters can be obtained by looking up related keywords (convolution, Gaussian, median, mean, erode, dilate, unsharp, etc.) on the Hypermedia Image Processing Reference index.

See also: Memory & Threads...

### 29.11.1 Convolve...

Does spatial convolution using a kernel entered into a text area.

A kernel is a matrix whose center corresponds to the source pixel and the other elements correspond to neighboring pixels. The destination pixel is calculated by multiplying each source pixel by its corresponding kernel coefficient and adding the results. If needed, the input image is effectively extended by duplicating edge pixels outward. There is no arbitrary limit to the size of the kernel but it must be square and have an odd width.

Rows in the text area must all have the same number of coefficients, the rows must be terminated with a carriage return, and the coefficients must be separated by one or more spaces. Kernels can be pasted into the text area using Ctrl+V.

Checking Normalize Kernel causes each coefficient to be divided by the sum of the coefficients, preserving image brightness.

The kernel shown is a $9 \times 9$ “Mexican hat”, which does both smoothing and edge detection in one operation. Note that kernels can be saved as a text file by clicking on the ‘Save’ button, displayed as an image using File⇒Import⇒Text Image…, scaled to a reasonable size using Image⇒Adjust⇒Size… and plotted using Analyze⇒Surface Plot…

See also: ConvolutionDemo macro
29.11.2 Gaussian Blur...

This filter uses convolution with a Gaussian function for smoothing [47].

\[ \sigma = \text{radius of decay to } e^{-0.5} \approx 61\% \]

Sigma is the radius of decay to \( e^{-0.5} \approx 61\% \), i.e., the standard deviation (\( \sigma \)) of the Gaussian (this is the same as in Adobe®Photoshop®, but different from ImageJ versions till 1.38q, in which radius was 2.5\( \times \sigma \) (cf. GaussianBlur.java).

Like all ImageJ convolution operations, it assumes that out-of-image pixels have a value equal to the nearest edge pixel. This gives higher weight to edge pixels than pixels inside the image, and higher weight to corner pixels than non-corner pixels at the edge. Thus, when smoothing with very high blur radius, the output will be dominated by the edge pixels and especially the corner pixels (in the extreme case, with a blur radius of e.g. 10\(^2\)0, the image will be replaced by the average of the four corner pixels).

For increased speed, except for small blur radii, the lines (rows or columns of the image) are downscaled before convolution and upscaled to their original length thereafter.

SEE ALSO: Noise – Add Noise, Accurate Gaussian Blur plugin, AnimatedGaussianBlur macro

29.11.3 Gaussian Blur 3D...

This command calculates a three dimensional (3D) gaussian lowpass filter using a 3-D Gaussian. It works with Stacks and Hyperstacks but not single-slice Color Composite Images. Refer to Gaussian Blur... for more information on sigma values.

SEE ALSO: Gaussian Blur 3D source code

29.11.4 Median...

Reduces noise in the active image by replacing each pixel with the median of the neighboring pixel values.

29.11.5 Mean...

Smooths the current image by replacing each pixel with the neighborhood mean.

29.11.6 Minimum...

This filter does grayscale erosion by replacing each pixel in the image with the smallest pixel value in that pixel's neighborhood.

SEE ALSO: Binary – Erode

29.11.7 Maximum...

This filter does grayscale dilation by replacing each pixel in the image with the largest pixel value in that pixel's neighborhood.

SEE ALSO: Binary – Dilate

29.11.8 Unsharp Mask...

Sharpens and enhances edges by subtracting a blurred version of the image (the unsharp mask) from the original. Unsharp masking subtracts a blurred copy of the image and rescales the image to obtain the same contrast of large (low-frequency) structures as in the input image. This is equivalent to adding a high-pass filtered image and thus sharpens the image.

**Radius**

The standard deviation (\( \sigma \) blur radius, cf. Gaussian Blur...) of the Gaussian blur that is subtracted. Increasing the Gaussian blur radius will increase contrast.

**Mask Weight**

Determines the strength of filtering, whereby \( \text{MaskWeight} = 1 \) would be an infinite weight of the high-pass filtered image that is added. Increasing the **Mask Weight** value will provide additional edge enhancement.

29.11.9 Variance...

Highlights edges in the image by replacing each pixel with the neighborhood variance.

29.11.10 Show Circular Masks...

Generates a stack containing examples of the circular masks used by the Median..., Mean..., Minimum..., Maximum..., and Variance... filters for various neighborhood sizes.

29.12 Batch

This submenu allows the execution of commands in a series of images without manual intervention.

**Batch** commands are non-recursive, i.e., they are applied to all the images of the chosen Input folder but not its subfolders. Nevertheless a directory hierarchy can be traversed using ImageJ macro language (cf. BatchProcessFolders macro).

Three critical aspects to keep in mind when performing batch operations that modify processed images:

- Files can be easily overwritten since the batch processor will silently override existing files with the same name.
- The destination **Output** folder should have adequate disk space to receive the created images.
- In the case of non-native formats, batch operations will be influenced by the behavior of the reader plugin or library (cf. Non–native Formats).
29.12.1 Measure...

This command measures all the images in a user-specified folder, by running the Analyze→Measure... command in all images of the chosen directory.

Note that measurements are performed on non thresholded images. In the case of TIFF images saved with active selections measurements are performed on the ROI and not the whole image.

See also: Analyze→Set Measurements..., Batch→Macro...

29.12.2 Convert...

Batch converts and/or resizes multiple images from a specified folder.

Input... Selects the source folder containing the images to be processed.

Output... Selects the destination folder where the processed images will be stored.

Output Format Specifies the output format that can be set to TIFF, 8-bit TIFF, JPEG, GIF, PNG, PGM, BMP, FITS, Text Image, ZIP or Raw (cf. Image Types and Formats and File→Save As→submenu).

Interpolation The resampling method to be used in case Scale Factor is not 1.00 (see Images→Size... and Image→Scale... [E]). For better results, Average when downsizing is automatically selected when scaling down images.

Scale Factor Specifies if images should be resized (see Image→Scale... [E]).

29.12.3 Macro...

Runs a macro over a specified folder. The last used macro is stored in the /ImageJ/macros/batchmacro.ijm file and remembered across restarts.

Input... Selects the source folder containing the images to be processed.

Output... Selects the destination folder where the processed images will be stored.

Output Format Specifies the output format that can be set to TIFF, 8-bit TIFF, JPEG, GIF, PNG, PGM, BMP, FITS, Text Image, ZIP or Raw (cf. Image Types and Formats and File→Save As→submenu).

Add Macro Code This drop-down menu contains macro snippets that can be combined to create the processing macro. Other statements can be pasted from the macro recorder or ImageJ’s editor while the dialog box is opened [38]. Previously written macros can be imported using Open... When editing the macro beware of any statements that may interfere with the normal operation of the batch processor (such as Close() or Open() calls).

Test Tests the macro on the first image of the Input... folder (the processed image will be displayed).

Open... Imports the macro on the first image of the Input... folder (the processed image will be displayed).

Save... Saves the assembled macro.

See also: Plugins→Macros→Record..., Batch→Virtual Stack...

29.12.4 Virtual Stack...

This command, that shares the same interface of Batch→Macro... (cf. BatchProcessor.java), allows virtual stack manipulations. E.g., Cropping a virtual stack can be performed by executing the following steps:

1. Open a virtual stack
2. Run Process→Batch→Virtual Stack...
3. Select an Output folder and Output format
4. Select ‘Crop’ from the Add Macro Code drop-down menu
5. Edit the macro code as needed and press the Test button to verify the macro
6. Click Process to create the cropped virtual stack

Note that cropped images are not loaded into memory but are saved to disk as they are cropped (see Virtual Stacks).

29.13 Image Calculator...

Performs arithmetic and logical operations between two images selected from popup menus described in the Image operations table. Image1 or both Image1 and Image2 can be stacks. If both are stacks, they must have the same number of slices. Image1 and Image2 do not have to be the same data type or the same size.

With 32-bit (float) images, pixels resulting from division by zero are set to NaN (Not a Number) if a zero pixel is divided by zero. The divide-by-zero value can be redefined in Edit→Options→Misc...
Operation Selects one of the thirteen available operators (see Image operations).

Create New Window If checked, a new image is created to hold the result. If unchecked, the result of the operation is applied directly to Image1.

32-bit (float) Result If checked, source images will be converted to 32-bit floating point before performing the operation.


Image Calculator... operations. On these examples source and destination images (8-bit grayscale) are displayed with inverted LUTs (White = 0; Black = 255) (cf. Lookup Tables submenu). Note that calculations between images can also be performed using copy and paste and the Edit> Paste Control... command.

<table>
<thead>
<tr>
<th>Source image (img1):</th>
<th>Destination image (img2):</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add: img1 = img1 + img2</td>
<td>Min: img1 = min(img1, img2)</td>
</tr>
<tr>
<td>Subtract: img1 = img1 - img2</td>
<td>Max: img1 = max(img1, img2)</td>
</tr>
<tr>
<td>Multiply: img1 = img1 x img2</td>
<td>Average: img1 = (img1 + img2)/2</td>
</tr>
<tr>
<td>Divide: img1 = img1 / img2</td>
<td>Difference: img1 =</td>
</tr>
<tr>
<td>AND: img1 = img1 AND img2</td>
<td>Copy: img1 = img2</td>
</tr>
<tr>
<td>OR: img1 = img1 OR img2</td>
<td>Transparent-zero</td>
</tr>
<tr>
<td>XOR: img1 = img1 XOR img2</td>
<td></td>
</tr>
</tbody>
</table>

See also: Calculator Plus plugin

29.14 Subtract Background...

Removes smooth continuous backgrounds from gels and other images [39]. Based on the concept of the ‘rolling ball’ algorithm described in Sternberg Stanley, Biomedical image processing, IEEE Computer, Jan 1983). Imagine that the 2D grayscale image has a third dimension (height) by the image value at every point in the image, creating a surface. A ball of given radius is rolled over the bottom side of this surface; the hull of the volume reachable by the ball is the background to be subtracted.

Rolling Ball Radius The radius of curvature of the paraboloid. As a rule of thumb, for 8-bit or RGB images it should be at least as large as the radius of the largest object in the image that is not part of the background. Larger values will also work unless the background of the image is too uneven. For 16-bit and 32-bit images with pixel value ranges different from 0–255, the radius should be inversely proportional to the pixel value range (e.g., for 16-bit images (pixel values 0–65535), typical values of the radius are around 0.2 to 5).

Light Background Allows the processing of images with bright background and dark objects.

Separate Colors (RGB images only) If unchecked, the operation will only affect the brightness, leaving the hue and saturation untouched.

Create Background (Don’t Subtract) If checked, the output is not the image with the background subtracted but rather the background itself. This option is useful for examining the background created (in conjunction with the Preview option). Create Background can be also used for custom background subtraction algorithms where the image is duplicated and filtered (e.g., removing ‘holes’ in the background) before creating the background and finally subtracting it with Process> Image Calculator... command.

Sliding Paraboloid If checked, the ‘rolling ball’ is replaced by a paraboloid that has the same curvature at the apex as a ball of that radius. This option allows any value of the radius>= 0.0001 (the ‘rolling ball’ algorithm requires a radius of at least 1). The ‘sliding paraboloid’ typically produces more reliable corrections since the ‘rolling ball’, a legacy algorithm (only kept for backward compatibility), is prone to edge artifacts.

To reduce the computing time the ‘rolling ball’ algorithm downscases the image in an inconsistent way. The ‘sliding paraboloid’ algorithm does not use downscaling and thus produces no downscaling artifacts. Nevertheless, the ‘sliding paraboloid’ is also an approximation, since it does not use a de facto paraboloid (an exact implementation would require a great computing effort) but it rather slides parabolae in different directions over the image.

Disable Smoothing For calculating the background (‘rolling the ball’), images are maximum-filtered (3x3 pixels) to remove outliers such as dust and then smoothed to reduce noise (average over (3x3 pixels). With Disable Smoothing checked, the unmodified image data
are used for creating the background. Check this option to make sure that the image data after subtraction will never be below the background.


**See also:** How to correct background illumination in brightfield microscopy by G. Landini, Auto Local Threshold, command’s source code, Rolling Ball Background Subtraction (the plugin that implemented this command in versions up to 1.39e)

### 29.15 Repeat Command [R]

Reruns the previous command. The **Edit**-**Undo** and **File**-**Open** commands are skipped.

**See also:** Undo and Redo

### 30 Analyze ⊿

This menu contains commands related to statistical measurements on image data, profile and histogram plotting and plugins related to image analysis.

#### 30.1 Measure... [m]

Based on the selection type, calculates and displays on the Results Table either area statistics, line lengths and angles, or point coordinates. Performed measurements can be specified in the **Set Measurements...** dialog box.

Area statistics are calculated for the complete image if there is no selection or for a selected subregion defined by one of the **Area Selection Tools**. For linear selections (Straight, Segmented and Freehand lines, see **Line Selection Tools**) length and angle (straight lines only) are also calculated. For Point selections (see **Point Tool** and **Multi-point Tool**), the **X** and **Y** coordinates are recorded. Note that **Measure...** [m] will paint (invasively) a mark over the measured point in foreground color unless **Mark Width** in the **Point Tool** options dialog box is set to zero.

With RGB images, results are calculated using brightness values. RGB pixels are converted to brightness values using the formula $value = \frac{red + green + blue}{3}$ or $value = 0.299 \times red + 0.587 \times green + 0.114 \times blue$ if **Weighted RGB Conversions** is checked in **Edit**-**Options**-**Conversions**.

Intensity statistics (Mean, Modal, Median, Min. & Max. Gray Value, Standard Deviation and Integrated Density) can be performed on area, line and multi-point selections. With lines, these parameters are calculated from the values of the pixels along the line (see **Plot Profile** [k]).

With area selections, the following parameters can measured: Area, Center of Mass, Centroid, Perimeter, Bounding Rectangle, Shape Descriptors, Fitted Ellipse, Feret’s Diameter, Skewness, Kurtosis and Area Fraction.

**See also:** Results Table, **Analyze Particles...**, Summarize, Distribution..., **Set Measurements...**, **Batch**-**Measure**...

#### 30.2 Analyze Particles... ⊿

This command counts and measures objects in binary or thresholded images (**Image**-**Adjust**-**Threshold**... [T] or Color Threshold...). Analysis is performed on the existing area selection or on the entire image if no selection is present.

It works by scanning the image or selection until it finds the edge of an object. It then outlines the object using the **Wand Tool**, measures it using the **Measure...** [m] command, fills it to make it invisible, then resumes scanning until it reaches the end of the image or selection. Press **Esc** to abort this process.

**Size**: Particles with size (area) outside the range specified in this field are ignored. Values may range between 0 and ‘Infinity’. For spatial scaled images (cf. **Set Scale**...) values are expressed in physical size square units or in pixels if **Pixel Units** is checked. Enter a single value and particles smaller than that value will be ignored.
Particle Analyzer (Analyze ➔ Analyze Particles...). Features of thresholded images can be extracted by specifying suitable Size and Circularity ranges and/or by choosing if particles should be traced by their outer edge or by flood filling (Include Holes checkbox).

Circularity  Particles with size circularity values outside the range specified in this field are also ignored. Circularly \((4\pi \times \frac{\text{Area}}{\text{Perimeter}^2})\) ranges from 0 (infinitely elongated polygon) to 1 (perfect circle).

Show  This drop-down menu specifies which image (or overlay) should ImageJ display after the analysis (see Display options of ParticleAnalyzer). Size, color and background of text labels can be adjusted in the Image ➔ Overlay ➔ Labels... prompt. Non-overlay outputs can be adjusted in a macro by using call("ij.plugin.filter.ParticleAnalyzer.setFontSize", size); and call("ij.plugin.filter.ParticleAnalyzer.setLineWidth", width); or call("ij.plugin.filter.ParticleAnalyzer.setLineCap", cap); or call("ij.plugin.filter.ParticleAnalyzer.setLineJoin", join); or ParticleAnalyzer.setLineDash(dash); or ParticleAnalyzer.setLineDashOffset(dashOffset); or ParticleAnalyzer.setLineDashPattern(pattern); or ParticleAnalyzer.setLineDashCap(cap); or ParticleAnalyzer.setLineDashJoin(join); in a script or plugin.

Nothing  Neither images nor Overlays will be displayed. Note that the particle analyzer will display a blank image when the count of detected particles is zero and Show is not Nothing.

Outlines  8-bit image containing numbered outlines of the measured particles (gray levels: Outlines:0; Labels:1; Background:255). If In situ Show is checked, the original image will be replaced by this image.

Bare Outlines  8-bit image containing simple outlines of the measured particles without labels (gray levels: Outlines:0; Background:255). If In situ Show is checked, the original image will be replaced by this image.

Masks  8-bit binary image containing filled outlines of the measured particles (gray levels: Masks:0; Background:255). If In situ Show is checked, the original image will be replaced by this image.

Ellipses  8-bit binary image containing the best fit ellipse (cf. Edit ➔ Selection ➔ Fit Ellipse) of each measured particle (gray levels: Ellipses:0; Background:255). If In situ Show is checked, the original image will be replaced by this image.

Count Masks  16-bit image containing filled outlines of the measured particles painted with a grayscale value corresponding to the particle number. If In situ Show is checked, the original image will be replaced by this image.

Overlay Outlines  Displays outlines of the measured particles in the image overlay, removing previously added Overlays.

Overlay Masks  Displays filled outlines of the measured particles in the image overlay, removing previously added Overlays.

Display Results  If checked, the measurements for each particle will be displayed in the Results Table.

Clear Results  If checked, any previous measurements listed in the Results Table will be cleared.

Summarize  If checked, the particle count, total particle area, average particle size, area fraction and the mean of all parameters listed in the Set Measurements... dialog box will be displayed in a separate Summary table. Note that while single images ‘Summaries’ are output to the same Summary table, stack Summaries are printed in dedicated tables (named Summary of [stack title]). Also, note that descriptive statistics on Results measurements can be obtained at any time using the Summarize command.

Add to Manager  If checked, the measured particles will be added to the ROI Manager...

Exclude on Edges  If checked, particles touching the edge of the image (or selection) will be ignored.

Include Holes  If checked, interior holes will be included. Disable this option to exclude interior holes and to measure particles enclosed by other particles. When this option is enabled, ImageJ finds the extent of each particle by tracing the outer edge. When it is disabled, ImageJ finds the extent by flood filling.
**Record Starts**  This option allows plugins and macros to recreate particle outlines using the doWand(x,y) macro function. The CircularParticles macro demonstrates how to use this feature.

**In situ Show**  If checked, the original image will be replaced by the binary mask specified in the Show drop-down menu. Note that this option does not apply to Overlay Outlines and Overlay Masks that are always displayed as non-destructive image Overlays on the measured image.


See also: Distribution Plotter, a macro that plots relative and cumulative frequencies on a double Y-axis graph

### 30.3 Summarize

For each column in the Results Table, calculates and displays the mean, standard deviation, minimum and maximum of the values in that column. This command is also available by right-clicking on the Results Table.

<table>
<thead>
<tr>
<th>Label</th>
<th>Area</th>
<th>Mean</th>
<th>Mode</th>
<th>Min</th>
<th>Max</th>
<th>Perim</th>
<th>Circ</th>
<th>Feret</th>
<th>IntDen</th>
<th>IntRen</th>
<th>Median</th>
<th>Sarea</th>
</tr>
</thead>
<tbody>
<tr>
<td>63</td>
<td>blobs.tif</td>
<td>5.3000</td>
<td>1.8842</td>
<td>1.3602</td>
<td>1.2802</td>
<td>2.4802</td>
<td>3.5566</td>
<td>5.2681</td>
<td>1.6031</td>
<td>9.9843</td>
<td>1.8402</td>
<td>1.0002</td>
</tr>
<tr>
<td>64</td>
<td>biobits.tif</td>
<td>4.9000</td>
<td>1.7262</td>
<td>1.7602</td>
<td>1.2802</td>
<td>2.2402</td>
<td>3.6761</td>
<td>4.5651</td>
<td>1.7031</td>
<td>8.4723</td>
<td>1.7602</td>
<td>1.0002</td>
</tr>
<tr>
<td>Mean</td>
<td>357.547</td>
<td>187.717</td>
<td>203.375</td>
<td>128</td>
<td>230.375</td>
<td>67.902</td>
<td>0.834</td>
<td>24.875</td>
<td>67786.258</td>
<td>0.500</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>SD</td>
<td>217.766</td>
<td>19.041</td>
<td>32.931</td>
<td>0</td>
<td>26.096</td>
<td>26.562</td>
<td>0.144</td>
<td>9.516</td>
<td>44012.25</td>
<td>32.111</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Min</td>
<td>1</td>
<td>128</td>
<td>128</td>
<td>128</td>
<td>128</td>
<td>2.282</td>
<td>0.406</td>
<td>1.414</td>
<td>128</td>
<td>128</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>Max</td>
<td>902</td>
<td>219.911</td>
<td>248</td>
<td>128</td>
<td>248</td>
<td>132.225</td>
<td>1</td>
<td>52.202</td>
<td>178720</td>
<td>240</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

See also: Analyze Particles..., Distribution...

### 30.4 Distribution...

Produces a relative frequency histogram from the data of a chosen column of the Results table.

Use the List or Copy buttons to save the histogram data. Mouse over the histogram bars to read the counts for each bin on the window’s lower right corner. Analyze> Histogram [h] describes in more detail ImageJ’s histogram window.

This command is also available by right-clicking on the Results Table.

**Parameter**  Specifies the parameter in the Results Table to be analyzed.

**Data points**  The number of rows that will be analyzed (informative).

**Automatic binning**  If checked, ImageJ will use the method described by David Scott to assess the optimal histogram bin width (see Scott DW, Optimal and data-based histograms. Biometrika, 66(3):605–610, Jan 1979).

If unchecked, the number of bins can be set with Specify bins and the starting and ending limits of the histogram with range.

See also: Color Picker Tool, XVII Embedding Color Annotations in Grayscale Images

### 30.5 Label

This command labels the active selection with the current measurement counter value, i.e., the number of rows present in the Results Table. Selection outline and label (at the selection centroid) are drawn invasively using current foreground/background colors. As for Edit>Draw [d], use the Edit>Options>Line Width... command, or double click on the line tool, to change the width of selection’s outline.

Selections can be labelled if they were previously analyzed (Analyze Particles... or Measure... [m] commands) and the parameter Centroid (cf. Set Measurements...) extracted in the Results table.

See also: Color Picker Tool, XVII Embedding Color Annotations in Grayscale Images

### 30.6 Clear Results

Erases the results table and resets the measurement counter. This command is also available by right-clicking on the Results Table.

### 30.7 Set Measurements...

Use this dialog box to specify which measurements are recorded by right-clicking on the Results Table. The dialog contains two groups of checkboxes: The first group controls the type of measurements (cf. Analyze Particles... or Measure... [m] commands) and the parameter Centroid (cf. Set Measurements...) extracted in the Results table.

The dialog contains two groups of check boxes: The first group controls the type of measurements that are printed to the Results table. The second group controls measurement settings. The eighteen checkboxes of the first group are:

**Area**  Area of selection in square pixels or in calibrated square units (e.g., mm$^2$, μm$^2$, etc.) if Analyze>Set Scale... was used to spatially calibrate the image.

**Mean gray value**  Average gray value within the selection. This is the sum of the gray values of all the pixels in the selection divided by the number of pixels. Reported in calibrated units (e.g., optical density) if Analyze> Calibrate... was used to calibrate the image. For RGB images, the mean is calculated by converting each pixel to grayscale using the formula $gray = (red + green + blue)/3$ or $gray = 0.299 \times red + 0.587 \times green + 0.114 \times blue$ if Weighted RGB Conversions is checked in Edit>Options>Conversions...
**Standard deviation** Standard deviation of the gray values used to generate the mean gray value. Uses the Results Table heading StdDev.

**Modal gray value** Most frequently occurring gray value within the selection. Corresponds to the highest peak in the histogram. Uses the heading Mode.

**Min & max gray level** Minimum and maximum gray values within the selection.

**Centroid** The center point of the selection. This is the average of the x and y coordinates of all of the pixels in the image or selection. Uses the X and Y headings.

**Center of mass** This is the brightness-weighted average of the x and y coordinates of all pixels in the image or selection. Uses the XM and YM headings. These coordinates are the first order spatial moments.

**Perimeter** The length of the outside boundary of the selection. Uses the heading Perim.. With IJ 1.44f and later, the perimeter of a composite selection is calculated by decomposing it into individual selections. Note that the composite perimeter and the sum of the individual perimeters may be different due to use of different calculation methods.

**Bounding rectangle** The smallest rectangle enclosing the selection. Uses the headings BX, BY, Width and Height, where BX and BY are the coordinates of the upper left corner of the rectangle.

**Fit ellipse** Fits an ellipse to the selection. Uses the headings Major, Minor and Angle. Major and Minor are the primary and secondary axes of the best fitting ellipse. Angle is the angle between the primary axis and a line parallel to the X-axis of the image. The coordinates of the center of the ellipse are displayed as X and Y if Centroid is checked. Note that ImageJ cannot calculate the major and minor axis lengths if Pixel Aspect Ratio in the Analyze>Set Scale... dialog is not 1.0. There are several ways to view the fitted ellipse:

1. The Edit>Selection>Fit Ellipse command replaces an area selection with the best fit ellipse.
2. The DrawEllipse macro draws (destructively) the best fit ellipse and the major and minor axis.
3. Select Ellipses from the Show: drop-down menu in the particle analyzer (Analyze>Particles...) and it will draw the ellipse for each particle in a separate window.

**Shape descriptors** Calculates and displays the following shape descriptors:

- **Circularity** $4\pi \times \frac{[\text{Area}]}{[\text{Perimeter}]}^2$ with a value of 1.0 indicating a perfect circle. As the value approaches 0.0, it indicates an increasingly elongated shape. Values may not be valid for very small particles. Uses the heading Circ.
- **Aspect ratio** The aspect ratio of the particle’s fitted ellipse, i.e., $\frac{[\text{Major Axis}]}{[\text{Minor Axis}]}$. If Fit Ellipse is selected the Major and Minor axis are displayed. Uses the heading AR.
- **Roundness** $4\pi \times \frac{[\text{Area}]}{[\text{Perimeter}]}$ or the inverse of Aspect Ratio. Uses the heading Round.
- **Solidity** $\frac{[\text{Area}]}{[\text{Convex Hull area}]}$. Note that the Edit>Selection>Convex Hull command makes an area selection convex.

**Feret’s diameter** The longest distance between any two points along the selection boundary, also known as maximum caliper. Uses the heading Feret. The angle (0–180 degrees) of the Feret’s diameter is displayed as Feret Angle, as well as the minimum caliper diameter (MinFeret). The starting coordinates of the Feret diameter (FeretX and FeretY) are also displayed (see also Feret’s Diameter macro and Chamfer distances and Geodesic diameters plugin).

**Integrated density** The sum of the values of the pixels in the image or selection. This is equivalent to the product of Area and Mean Gray Value. With IJ 1.44c and later, Raw integrated density (sum of pixel values) is displayed under the heading RawIntDen when Integrated density is enabled. The Dot Blot Analysis tutorial demonstrates how to use this option to analyze a dot blot assay.

**Median** The median value of the pixels in the image or selection.

**Skewness** The third order moment about the mean. The documentation for the Moment Calculator plugin explains how to interpret spatial moments. Uses the heading Skew.

**Kurtosis** The fourth order moment about the mean. Uses the heading Kurt.

**Area fraction** For thresholded images is the percentage of pixels in the image or selection that have been highlighted in red using Image>Adjust>Threshold... [T]. For non-thresholded images is the percentage of non-zero pixels. Uses the heading %Area.

**Stack position** The position (slice, channel and frame) in the stack or hyperstack of the selection. Uses the headings Slice, Ch and Frame.

N.B.: For line selections the heading Length is created. For straight line selections, Angle is recorded even if Fit Ellipse is unchecked. Also, note that measurements that do not apply to certain selection types may be listed as NaN, Infinity or −Infinity.

The second part of the dialog controls measurement settings:

**Limit to threshold** If checked, only thresholded pixels are included in measurement calculations. Use Image>Adjust>Threshold... [T] to set the threshold limits. This setting affects only thresholded images (see Settings and Preferences).

**Display label** If checked, the image name and slice number (for stacks) are recorded in the first column of the Results Table, e.g., nii-stack.tif. For renamed selections (Edit>Selection>Properties... [P]) or selections measured via ROI Manager’s measure command (see ROI Manager...), the selection label is appended, e.g., blobs.tif0339-0163 or blobs.tif0339-0163 Selection.

**Invert Y coordinates** If checked, the XY origin is assumed to be the lower left corner of the image window instead of the upper left corner (see also Image>Properties... [P]).

**Scientific notation** If checked, measurements are displayed in scientific notation, e.g., 1.48E2.

**Add to Overlay** If checked, measured ROIs are automatically added to the image overlay (see overlays). Appearance of overlay selections can be adjusted using Image>Overlay>Overlay Options... Labels...

**Redirect to** The Image selected from this popup menu will be used as the target for statistical calculations done by Analyze>Measure... [M] and Analyze>Particles... commands. This feature allows you to outline a structure on one image and measure the intensity of the corresponding region in another image.

**Decimal places** This is the number of digits to the right of the decimal point in real numbers displayed in the Results Table and in Histogram windows (Analyze>Histogram [h]).
30.8 Set Scale...

Use this dialog to define the spatial scale of the active image so measurement results can be presented in calibrated units, such as \( \text{mm} \) or \( \mu \text{m} \).

Before using this tool, use the straight line selection tool to make a line selection that corresponds to a known distance. Then, bring up the Set Scale... dialog, enter the Known Distance and unit of measurement, then click ‘OK’. The Distance in Pixels field will be automatically filled in based on the length of the line selection.

As described in Image Properties...[P], \( \mu \) and \( \text{Å} \) symbols can be typed using \( \text{Alt} \) and \( \text{Alt Shift} \) \( A \), respectively. \( \mu \text{m} \) can also be defined as ‘\( \text{mm} \)’, or ‘\( \text{micron} \)’.

Setting Pixel Aspect Ratio to a value other than 1.0 enables support for different horizontal and vertical spatial scales, e.g., 100\( \mu \text{m} \)/cm horizontally and 95\( \mu \text{m} \)/cm vertically. To set the Pixel Aspect Ratio:

1. Measure the width and height (in pixels) of a digitized object with a known 1:1 aspect ratio.
2. Enter the measured width (in pixels) in Distance in Pixels. Enter the known width in Known Distance.
3. Calculate the aspect ratio by dividing the width by the height and enter it in Pixel Aspect Ratio.

When Global is checked, the scale defined in this dialog is used for all opened images during the current session instead of just the active image, see XXIII Global Calibrations.

Click to Remove Scale resets Distance in Pixels field and Known Distance to zero and the Unit of Length to ‘pixel’.

See also: Tutorials showing how to use this command: Examples of Image Analysis Using ImageJ by Larry Reinking, Measuring DNA Contour Lengths and Spatial Calibration (Fiji). The Copy_Pixel_Size plugin

30.9 Calibrate...

Use this dialog box to calibrate an image to a set of density standards, for example radioactive isotope standards or a calibrated optical density step tablet. Note that, in general, calibrations cannot be applied to 32-bit images (the pixel intensity unit of 32-bit images can still be changed, nevertheless).

The calibration procedure is done in three steps:

1. Use Analyze▷Clear Results to reset the measurement counter, use one of the Area Selection Tools and Analyze▷Measure...[M] to record the mean gray value of each of the standards.
2. When finished making the measurements, select Analyze▷Calibrate... to display the Calibrate... dialog box. To calibrate the image, enter the known standard values in the right column. The left column will be already populated with the measured mean gray values. Select a curve fitting method from the popup menu, enter the unit of measurement, and click ‘OK’. If Show plot is checked, ImageJ will then display the calibration function on a separate window. Note that both columns must contain the same number of values.
3. If the calibration function is not satisfactory, bring up the Calibrate... dialog box again and select a different curve fitting method.

In addition to the functions that can be chosen from the drop-down menu (described in Curve-Fitter’s built-in functions) two other functions are available that do not require any measurement of OD standards:

- **Uncalibrated OD** As mentioned in Gels▷Gel Analyzer Options... causes ImageJ to convert gray values from 8-bit images to uncalibrated optical density values using the function \( \text{Unc. OD} = \log_{10}(255/\text{Pixel value}) \). This conversion can only be performed on 8-bit images.

- **Pixel Inverter** Linear function defined by \( \text{Inverted pixel} = \text{Bit-depth} – 1 – \text{Pixel value} \), with Bit-depth being 255 for 8-bit images or 65535 for 16-bit images.

See also: Optical Density Calibration tutorial, Global Calibrations, Calibration Bar..., Curve Fitting..., Image Types and Formats

30.10 Histogram [h]

Calculates and displays a histogram of the distribution of gray values in the active image or selection.

The X-axis represents the possible gray values and the Y-axis shows the number of pixels found for each gray value. The horizontal LUT bar below the X-axis is scaled to reflect the display

---

\( ^* \)This shortcut is shown on Windows and Linux but not on Mac OSX as it conflicts with the system wide ‘Hide’ shortcut. However, the \( \text{Ctrl} \) shortcut (without holding down \( \text{Cmd} \)) does work on OSX.
**XXIII Global Calibrations**

Calibration settings related to spatial (pixel width, height and voxel depth), temporal (frame interval) and luminance (brightness) information can be set globally, i.e., can be applied to all images opened during the current session instead of just the active image. The Global flag can be set in three dialog prompts:

1. **Image>Properties...** (pixel width, height, voxel depth, frame interval)
2. **Analyze>Set Scale...** (pixel width and height)
3. **Analyze>Calibrate...** (pixel intensity)

Once Global calibration is set, a ‘(G)’ is displayed in all image titles until ImageJ is closed. A warning message is displayed when a calibrated image with conflicting calibration is opened and the Global option is enabled.

Choose Disable Global Calibration to stop using global settings or Disable These Messages to keep respecting global settings, ignoring the calibration of the newly opened image.

Use **min/max**. If checked, the X-axis range is determined by the minimum and maximum values in the image or selection. If unchecked, X Min and X Max values can be specified to fix the X-axis range.

Y Max Fixes the Y-axis range. Type 'Auto' to have the range determined by the largest bin count.

The `getHistogram()` and `Plot.getValues()` macro functions can be used to get the 'Value' and 'Count' data displayed when you click the List button (cf. (8) Obtaining Histogram Lists).

### 30.11 Plot Profile [k]

Displays a two-dimensional graph of the intensities of pixels along a line or rectangular selection. The X-axis represents distance along the line and the Y-axis is the pixel intensity. NaN values in 32-bit images (see Image Types and Formats) are ignored.

For rectangular selections or line selections wider than one pixel, displays a 'column average plot', where the X-axis represents the horizontal distance along the selection and the Y-axis the vertically averaged pixel intensity.

To average horizontally, hold **Alt** or check Vertical Profile in Edit>Options>Profile Plot Options... For real-time examinations, activate Live mode to continuously update the profile as the selection is moved or resized.

To obtain profiles of several selections in a single plot use the ROI Manager’s Multi Plot command (**Analyze>Tools**>ROI Manager...). Other types of area selections such as oval or freehand ROIs can be profiled by first running Edit>Selection>Area to Line, which will convert these ROIs to line selections.

Use the List, Save... or Copy... buttons to view and save the profile data. Use Edit>Options>Profile Plot Options... to adjust how plots are generated.

**See also:** Rectangular Selection Tool, Line Selection Tools, Plot Z-axis Profile... Surface Plot... Dynamic Profiler, Oval Profile Plot, Radial Profile Plot, Radial Profile Extended plugins, StackProfilePlot macro.

---

**Range of the Image** [41]. The total pixel Count is also calculated and displayed, as well as the Mean, standard deviation (StdDev), minimum (Min), maximum (Max) and modal (Mode) gray value.

Click on Live to monitor the histogram while browsing stacks or while moving a ROI. Value / Count pairs (i.e., grayscale value corresponding to the X-axis cursor position / the number of pixels that have that intensity) are displayed on the bottom right while mousing over the histogram window.

With RGB images, the default histogram is calculated by converting each pixel to grayscale using the formula gray = (red + green + blue)/3 or gray = 0.299 × red + 0.587 × green + 0.114 × blue if Weighted RGB Conversions is checked in Edit>Options>Conversions... However, single-channel RGB histograms can be obtained by repetitively clicking on the RGB button.

Use the List or Copy buttons to save the histogram data. Click on Log to display a log-scaled version of the histogram (overlaid in gray).

With 16-bit images, the range of gray values between the Min and Max values is divided into 256 bins. With 32-bit images, the number of bins is specified in the depicted dialog box. With any image type, an options dialog can be called with **Alt** or by holding **Alt** while clicking on Analyze>Histogram [h].

**Bins** Specifies the number of bins.
30.12 Surface Plot...

Displays a three-dimensional graph of the intensities of pixels in a grayscale or pseudo color image (non-RGB images). The plot is based on the existing rectangular selection or on the entire image if no selection is present. A stack of plots can be produced when the source image is a stack or hyperstack. In this case, closing the plot stack window will abort the plotting process.

**Polygon Multiplier** Adjusts the number of profiles used to generate the plot.

**Draw Wireframe** If checked, the outline of each profile will be drawn in black.

**Shade** If checked, a shaded plot will be generated using the LUT of source image.

**Draw Axis** If checked, the three axis will be drawn and labelled.

**Source Background is Lighter** If checked, lighter areas in the source image will represent lower elevations (valleys) while darker areas in the source image will represent higher elevations (peaks).

**Fill Plot Background with Black** If checked, the plot is drawn with a black background, otherwise white will be used.

**One Polygon Per Line** If checked, all polygons will be drawn.

**Smooth** If checked, sharp fluctuations will be smoothed. Note that some plots can be further improved by adjusting the contrast of the source image or smoothing it.

**See also:** Plot Profile [k], Interactive 3D Surface Plot plugin (it works with all image types and viewing angle, perspective, scale, lighting and smoothing can be interactively adjusted), 3D Color Inspector/Color Histogram

**XXIV Using Scanners in Densitometry**

Electrophoretic gels such as Western blots need frequently to be quantified in order to translate biochemical results into statistical values (see Gels>). Independently of the measurement method, you should be familiar with the Beer–Lambert law, so that you are aware that the optical density (absorbance) of the staining to be measured must be proportional to the concentration of the probed material. An obvious corollary of the Beer–Lambert law is that saturated stains (or overexposed films) cannot be quantified.

Generally speaking, office scanners are not suitable for densitometric analysis. That being said, and in the absence of better alternatives, they can be used to digitize gels and X-ray films if the following guidelines are respected:

**Samples must be scanned in transmission mode** Most flatbed scanners have the light source and detector located on the same side of the instrument (reflection mode). Under these conditions, light interacts with semi-transparent objects in a complex manner with reflected light re-passing through the object on its way to the detector. In addition, thick samples (such as Coomassie-stained gels) are imaged from the surface under reflection light.

Scanners can only serve as densitometers if the light source and detector are on opposite sides of the instrument (transmission mode). This is critical for quantitation of absorbances and requires an adapter for transparencies or negatives to be mounted or built-in on the scanner.

**The scanner must offer a linear intensity response** The response characteristics of most scanner detectors is unknown, therefore the film/gel should always be scanned next to a photographic gray scale image of known optical intensity (optical tablet or step wedge, see optical density calibration tutorial). Autogain must be disabled during digitization to ensure generally linear intensity responses.

In addition, images should also be scanned with sufficiently high spatial and dynamic (bit-depth) resolutions (the optical—not interpolated—resolution of a scanner can also be measured: e.g., high contrast patterns 84.6 µ (1/360 inch) apart should be distinguishable on images scanned at 300 dpi). Note that, in general, a posteriori background corrections on scanned images should also be avoided.

All these issues have been discussed in detail in Gassmann et al. “Quantifying Western blots: pitfalls of densitometry” [38] (N.B.: The paper benchmarks a legacy version of Process>Subtract Background…).

30.13 Gels>

Use the commands in this submenu to analyze one-dimensional electrophoretic gels. Electrophoretic densitometry is discussed in more detail in XXIV Using Scanners in Densitometry. These commands use a simple graphical method that involves generating lane profile plots, drawing lines to enclose peaks of interest, and then measuring peak areas (i.e., definite integrals) using the Wand Tool…. Note that this technique cannot be used to compare bands on different gels unless gels are calibrated to known standards.

The commands listed in this submenu are:

**Select First Lane 1** Requires a rectangular selection. Note that lanes are assumed to be vertical unless the width of the initial selection is at least twice its height.
Select Next Lane \(2\) To be used after the first rectangular ROI is moved over the adjacent
lanes. Note that all selections must have the same dimensions.

Plot Lanes \(3\) Generates the lane profile plots. ImageJ assumes that only one plot is created
per analysis. As a consequence, re-running this command more than once within the same
analysis will cause an error message: “You must first use the ‘Select First Lane’ command”.
To recreate plotted profiles use the Re-plot Lanes command.

Re-plot Lanes Recreates the lane profile plots. If Plot Lanes has not yet been run an error
message is displayed: “The data needed to re-plot the lanes is not available”.

Reset Resets the analysis.

Label Peaks Uses the area measurements obtained with the Wand Tool to label lane peaks
(see Gel Analyzer Options...).

Gel Analyzer Options... Use this dialog to control the behavior of the gel analyzer.

Vertical / Horizontal scale factor Specifies the scale factor at which the lane profile
plots are displayed.

Uncalibrated OD If checked, ImageJ will convert gray values to uncalibrated optical
density values. As explained in Analyze Calibrate..., ImageJ converts pixel intensities
into optical density using the function: Unc. OD = log\(_{10}\)\(\times\)255/pixel value. As mentioned
earlier, the conversion can only be performed on 8-bit images. Thus, when dealing
with higher bit-depth images (see Image Types and Formats), the gel analyzer works
on a 8-bit copy of the gel (kept hidden from the user) when using this option.

Label With Percentages If checked, the Label Peaks command will print to the Results
Table the peak percentage and use it to label the plot. The percentage value is
obtained by dividing the area of each peak by the sum of all measured peaks from all
lanes.

Invert Peaks If checked, peaks will be inverted, i.e, bands darker than background will
have positive peaks, bands lighter than background will have negative peaks. This
setting does not change the analysis (see Image Lookup Tables→Invert LUT).

For practice, refer to the video tutorial on the ImageJ wikipage and use the File→Open Samples→
Gel sample image (1-D gel) to perform the following steps. Note that a copy of the gel image
with the lane outlines can be created at any point using the Image→Overlay→Flatten [F] command.

1. Use the rectangular selection tool to outline the first lane. This should be the left most
lane if the lanes are vertical or the top lane if the lanes are horizontal.

2. Select Gel→Select First Lane \(1\) and the lane will be outlined and ‘Lane 1 selected’
displayed in the status bar.

3. Move the rectangular selection right to the next lane (or down if the lanes are horizontal)
and select Gel→Select Next Lane \(2\). The selected lane is outlined and labelled, and
‘Lane n selected’ is displayed in the status bar.

4. Repeat the previous step for each remaining lane.

5. Select Gel→Plot Lanes \(3\) to generate the lane profile plots.

6. Use the Straight Line Selection Tool to draw base lines and/or drop lines so that each peak
of interest defines a closed area (ImageJ will automatically switch to the Straight Line
tool). Note that you can hold Shift to constrain lines to be either horizontal or vertical.
To access to all the lanes, it may be necessary to scroll the image vertically using the
Scrolling Tool (Hold down the space bar to temporarily switch to this tool).

7. For each peak, measure the size by clicking inside the peak with the Wand Tool. If
necessary, scroll the image vertically by holding down the space bar and dragging.

8. Select Gel→Label Peaks to label each measured peak with its size as a percent of the total
size of the measured peaks.

See also: Calibrate..., SinglePanelGelAnalyzer macro, Video tutorial on ImageJ wikipage, Luke
Miller’s tutorial, Dot Blot Analysis, Dot Blot Analyzer tooset

30.14 Tools

This submenu provides access to various image analysis plugins.

30.14.1 Save XY Coordinates

Writes to a text file the XY coordinates and pixel value of all non-background pixels in the
active image. Background is assumed to be the value of the pixel at the upper left corner of
the image. For grayscale images, writes three values per line (x, y, and value), separated by
spaces. For RGB images, writes five values per line (x, y, red, green and blue). The origin of
the coordinate system is at the lower left corner of the image.

The number and percentage of non-background pixels is printed to the Log Window if Suppress
Log output is not checked.

See also: Edit→Selection→Properties...
30.14.2 Fractal Box Count...

Estimates the fractal dimension \(D\) of a binary image. \(D\) can be used as a measure of pattern complexity (cell shape, vascularization, textures, etc.) and is specially relevant in cases in which Euclidean measures such as diameter or length are not good descriptors of complexity.

The command counts the number of boxes of an increasing size needed to cover a one pixel binary object boundary and implements the method described in T. G. Smith, Jr., G. D. Lange and W. B. Marks, Fractal Methods and Results in Cellular Morphology, *J Neurosci Methods*, 69:1123–126, 1996.

A plot is generated with the log of size on the X-axis and the log of count on the Y-axis and the data is fitted with a straight line. The slope \(S\) of the line is the negative of the fractal dimension, i.e., \(D = -S\). ‘Size’ \((S)\) and ‘count’ \((C)\) are printed to the Results Table. Refer to the source code for additional information.

See also: Fractal Dimension and Lacunarity plugin

30.14.3 Analyze Line Graph

This command uses the Particle Analyzer to extract sets of coordinate data from digitized line graphs. The following procedure describes how to use it:

1. Open the image containing the graph. For practice, use the File▷ Open Samples▷ Line Graph (21K) sample image. Make sure your graph is a grayscale image (Image▷ Type▷ 8-bit). Analyze Line Graph will assume that the graph is displayed on a white background so images with darker backgrounds must be adjusted beforehand (see e.g., Edit▷ Invert [I] and Process▷ Binary▷ Make Binary).

2. Set background color to white using the Color Picker Tool or the Color Picker window. Use any of the Area Selection Tools as an eraser (press Backspace) to erase in order to isolate the single graph curve to be measured. Alternatively, use one of the drawing tools (Pencil or Brush) to paint directly in background color.

3. Open the Threshold... [T] tool (Shift [T]) and adjust the threshold levels so that the curve is highlighted in red.

4. Select the curve with the Wand Tool and run Edit▷ Clear Outside to erase everything on the canvas but the curve.

5. With the line still highlighted by the threshold widget, run Analyze▷ Tools▷ Analyze Line Graph to get the XY coordinates of the traced line (you can hold down [Alt] while selecting the command to reveal the actual image that is processed).

6. On the newly obtained plot, select List, Copy or Save (these commands are described in Plot Profile [K]) to export the curve coordinates into a spreadsheet application.

The exported values are tabulated in pixel coordinates, unless the digitized graph has been spatially calibrated using Analyze▷ Set Scale... or Image▷ Properties... [P].

See also: Figure Calibration, a simple plugin to retrieve data from other types of graphs such as scatter plots and histograms

30.14.4 Curve Fitting...

ImageJ’s CurveFitter provides a simple tool for fitting various functions to X- and Y-data, using an improved multithreaded simplex algorithm [42]. This strategy (an iterative method) is a kind of “trial and error” procedure in which the parameters of the fitting model are adjusted in a systematic way until the equation fits the data as close as required. It proceeds by: 1) Making first guesses of all the non-linear parameters; 2) Computing the model, comparing it to the data set and calculating a fitting error; 3) If the fitting error is large, the CurveFitter will systematically change parameters and return to step 2). The loop stops when the fitting accuracy is met, which in difficult cases may never happen. In the latter case, the procedure terminates after an imposed number of iterations or restarts.

The typical usage of this command is listed below:

1. Tabular data is entered or copied in the input window or alternatively, a two column text file is opened by clicking on the Open button. Values may be separated by spaces, tabs, commas or semicolons.

2. The function to be fit is selected from the drop-down menu. Several built-in functions are available (see CurveFitter’s built-in functions). User defined functions with up to six parameters are also possible by choosing *User-defined*. Note that elimination of parameters by linear regression does not take place for user-defined functions. As a consequence, the performance of custom functions does not fully match that of built-in functions.

3. Once the Fit button is pressed, ImageJ displays a graph of the data with the fitted curve as depicted in CurveFitter. If Show Settings is checked, detailed information about the fit (including measures of goodness of fit) is printed to the Log Window and the user is prompted to re-adjust the simplex fitting options, namely:

   **Maximum number of iterations** The number of maximum iterations in which the CurveFitter will try to improve upon the parameter values to get the best fit. Usually the algorithm reaches optimal convergence before reaching the default value.
**Number of restarts** To ensure that the result is trustworthy (i.e., that it did that it did not get “stack” or find a local minimum). CurveFitter tries at least twice to find the minimum, with different starting points. If the two results are not the same, *Number of restarts* determines how often it may start two additional runs, until the best two results agree within the error tolerance. There is no limit for the number of restarts, apart from the maximum number of iterations.

4. Click **Apply** to create a 32-bit copy (see Image Types and Formats) of the current image transformed with the chosen function.

**CurveFitter’s built-in functions.** A more detailed summary of the CurveFitter’s abilities is available online. The complete documentation can be accessed through the ImageJ API. Most functions use linear regression to determine the first two coefficients directly, which generally increases the quality of the result [12]. As explained in Curve Fitting..., simple fitting options can be adjusted by selecting the Show settings checkbox. For polynomials, the proposed criteria are known to yield optimal convergence.

<table>
<thead>
<tr>
<th>Function</th>
<th>Formula</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Straight line</td>
<td>$y = a + bx$</td>
<td>cf. Pixel Inverter (Analyze&gt;Calibrate…)</td>
</tr>
<tr>
<td>2nd-8th degree polynomial</td>
<td>$y = a + bx + cx^2 + \ldots + iz^8$</td>
<td>The range of $x$ values should not be too far from 0, especially for higher-order polynomials</td>
</tr>
<tr>
<td>Power†</td>
<td>$y = a x^b$</td>
<td>Optionally, fit can be performed without linear regression during minimization. Most curve-fitting programs such as Microsoft® Excel® use regression. Fitting without regression assumes equal weight for all data points and is insensitive to zero or negative data</td>
</tr>
<tr>
<td>Exponential‡</td>
<td>$y = a e^{cx}$</td>
<td></td>
</tr>
<tr>
<td>Exponential with offset</td>
<td>$y = a e^{-(hx)} + c$</td>
<td></td>
</tr>
<tr>
<td>Exponential recovery</td>
<td>$y = a (1 - e^{-(hx)}) + c$</td>
<td>cf. FRAP_Profiler plugin</td>
</tr>
<tr>
<td>Log</td>
<td>$y = a + b \ln(x)$</td>
<td></td>
</tr>
<tr>
<td>Rodbard (NIH Image)</td>
<td>$y = \frac{d}{c}(\frac{x-b}{a})^{\frac{1}{c}}$</td>
<td>[12], see DeLean A, Musson PJ, Rodbard D. Simultaneous analysis of families of sigmoidal curves: application to bioassay, radioligand assay, and physiological dose-response curves. Am J Physiol. 1978 Aug;235(2):E97-E102 [PMID: 686171]</td>
</tr>
<tr>
<td>Rodbard</td>
<td>$x = \frac{d}{c}(\frac{y-b}{a})^{c}$</td>
<td></td>
</tr>
<tr>
<td>Inverse</td>
<td>$y = c \times (\frac{d-x}{e})^{\frac{1}{h}}$</td>
<td></td>
</tr>
<tr>
<td>Gaussian</td>
<td>$y = a + (b-a) \times e^{-\frac{(x-c)^2}{2\sigma^2}}$</td>
<td></td>
</tr>
<tr>
<td>Gamma variate</td>
<td>$y = b \times (x-a)^{\frac{1}{b}} \times e^{-\frac{(x-a)^2}{2\sigma^2}}$</td>
<td></td>
</tr>
</tbody>
</table>

See also: [CurveFittingDemo], [RodbardSigmoidFit] and [PlotSigmoidDerivatives] macros, Profile Plot Options…

---

30.14.5 **ROI Manager**

The ROI (Region of Interest) Manager is a tool for working with multiple selections. Selections can be from different locations on an image, from different slices of a stack or from different images. All selection types, including points, lines and text, are supported.

**Add** Click Add to add the current selection to the list, or press Shift A, the keyboard shortcut for the Edit>Selection>Add to Manager [t] command. The ROI manager creates a three part label. The first part (stacks only) is the slice number, the second is the Y-coordinate of the selection and the third is the X-coordinate. Click on a label to restore the associated selection to the current image. With stacks, the selection is restored to the slice it came from. Hold down Shift while clicking Add to ‘Add and Draw’. Hold down Alt while clicking Add to ‘Add and Rename’. Install the ROIManager Macros macro set and you will be able to add a selection by pressing 1, ‘add and rename’ by pressing 2, ‘add and draw’ by pressing 3 and ‘add and advance to the next slice’ by pressing 4.

**Update** Replaces the selected ROI on the list with the current selection, updating the x/y-position of the ROI in Stacks and Hyperstacks.

**Delete** Deletes the selected ROIs from the list. Deletes all ROIs if none is selected.

**Rename** Renames the selected ROI. The chosen string will be used as label (Labels checkbox) if Use ROI names as labels is checked in the More>Options… dialog. The selected ROI can also be renamed using the Properties… button. Note that while it is not possible to rename multiple ROIs simultaneously, you can use ROI Manager Tools to rename multiple ROIs.

**Measure** Measures the selected ROIs, or if none is selected, all ROIs on the list. Use Analyze> Set Measurements… to specify the measuring parameters.

**Deselect** Deselects any selected ROIs on the list. As mentioned in XXV Selecting ROIs in the ROI Manager, when items are deselected subsequent ROI Manager commands are applied to all ROIs.

**Properties**… Similarly to Edit>Selection>Properties… [y], opens a dialog box in which is possible to assign a contour color (Stroke color) of a certain Width or a Filling color. Set
Toggles the display of all ROI Manager Overlays. If 
_changed in
.IJ
improved
.New in
... can later be retrieved using _image>Overlay>From ROI Manager. When saving the image as tiff, Overlays are stored in the TIFF header and can later be retrieved using _image>Overlay>From ROI Manager.

Labels Toggles overlay labels displayed by _Show All_. Labels are customized using _More>Labels…, a shortcut to _image>Overlay>Labels…_.

More Toggles the display of all ROI Manager Overlays. If _Labels_ is active, ROIs will also be labelled. Once _Show All_ is checked, ROIs can be re-activated by Alt-clicking, Control-clicking or long-pressing (_1/4 second or longer_). Re-activated selections that are moved or edited are automatically updated.

When _Show All_ is active and the ROI Manager window is closed, a dialog box is displayed providing the option to save the displayed ROIs by moving them to the image overlay by running _image>Overlay>From ROI Manager_. When saving the image as tiff, Overlays are stored in the TIFF header and can later be retrieved using _image>Overlay>To ROI Manager._

Open… Opens a _roi_ file and adds it to the list or opens a ZIP archive (_zip_ file) and adds all the selections contained in it to the list.

Save… Saves the selected ROI as a _roi_ file. If no ROIs are selected, saves all the ROI Manager selections in a ZIP archive.

**Draw** Alias for Edit>Fill [f].

**AND** Uses the conjunction operator on the selected ROIs to create a composite selection. **OR (Combine)** Uses the union operator on the selected ROIs to create a composite selection. Combines all ROIs if none is selected.

**XOR** Uses the exclusive or operator on the selected ROIs to create a composite selection. All ROIs are considered if none is selected.

**Split** Splits the current selection (it must be a composite selection) into its component parts and adds them to the ROI Manager.

**Add Particles** Adds objects segmented by the particle analyzer to the ROI Manager. Requires that **Record Starts** be checked in the _Analyze>Add Analyze Particles…_ dialog box. Particle analyzer objects can also be added to the ROI Manager by checking **Add to Manager** in the Analyze Particles dialog box.

Multi Measure Measures all the ROIs on all slices in the stack, creating a Results Table with either one row per slice (_One Row Per Slice_ is checked in the dialog) or one row per measurement (_Multi Plot_).

Sort Sorts the list in alphabetical order.

Specify… Alias for the Edit>Selection>Specify… prompt that allows the creation of area ROIs at specific locations.

Remove Slice Info Removes the information in the ROI names that associates them with particular stack slices (see ROI Manager illustration).

Help Opens _http://imagej.nih.gov/ij/docs/menus/analyze.html#manager_.

Labels… Alias for _image>Overlay>Labels…_, which allow the customization of selection labels when _Show All_ is active.

List Prints a table detailing the properties of the ROIs stored in the Manager, including: **Index** (cf. _roimanager("index")_ macro function), **Name**, **XY coordinates** of ROI center (pixels), and **stroke Color**.

Options… Displays a dialog box (depicted in ROI Manager) that allows you to set several ROI Manager settings:

**Associate "Show All" ROIs with slices** If checked, _Show All_ will only reveal ROIs when browsing their respective slice. If unchecked, ROIs are shown in all stack slices.

**Restore ROIs centered** If checked, ROIs opened by _More>Open…_ are centered on the image canvas. This option avoids loaded ROIs to be displayed out of boundaries when the image has been resized.

**Use ROI names as labels** If checked, ROI names are used as selection labels when in the **Labels** checkbox is active. If unchecked, the ROI position in the Manager’s list is used. Selections can be renamed using either **Rename** or **Properties…**
XXV Selecting ROIs in the ROI Manager

For most ROI Manager operations Deselect works as a Select All button, e.g., to measure all ROIs in manager one would press Deselect then Measure. In addition, it is possible to select contiguous ROIs in the ROI Manager list with a single Shift–click. Non-contiguous ROIs can be selected by Control–click (Command–click on Mac OSX).

When Show All is active, Overlays that are not stored in the ROI Manager will not be re-activated by Alt-clicking, Control-clicking or long-pressing (9/4 second or longer). This is a reminder that those overlay selections (added to the image overlay via Image ⊿ Overlay ⊿ Add Selection... [b]) are not under the control of the ROI Manager. You will be able to activate them as soon as Show All is unchecked.

SEE ALSO: ROI Manager Tools, ⊿ROIManagerMacros, ROI Color Coder, Edit ⊿ Selection, ⊿Image ⊿ Overlay ⊿, ⊿Analyze ⊿ Tools ⊿ Synchronized windows, XIX Hexadecimal Color Values, XX Transferring Selections Between Images

30.14.6 Scale Bar...

Draws a labelled spatial calibration bar.

Location Adjusts the position of the calibration bar. If there is a selection, the bar is initially drawn at the selection.

Width Length of the bar in calibrated units.

Height Height of the bar in pixels.

Font Size Adjusts the font size of the scale bar label.

Color Adjusts the text color (see XVII Embedding Color Annotations in Grayscale Images).

Background Adjusts the filling color of the label text box.

The Calibration Bar Macros can be used to add a calibration bar to a stack or to all the images and stacks in a folder.

SEE ALSO: Calibrate..., XXIII Global Calibrations

30.14.8 Synchronize Windows

Synchronizes mouse motion and input between multiple windows so that a ROI drawn in one image is replicated in all other Synchronized windows [46]. A synchronization cursor indicates the location of the mouse across the synchronized window set.

Synchronized window set Images to be synchronized are specified in this list, containing all open images. To select a consecutive group of images, click the first item, press and hold down Shift, and then click the last item. Alternatively, click on the first item and drag it across. To select non-consecutive images, press and hold down Ctrl, then click each item to be selected. Use the Un/Synchronize All buttons to de/select all listed images.

Sync cursor If checked, the cursor is synchronized across selected images, with the mouse pointer changing to a red X-shape cursor. When unchecked, mouse movements become restricted to the active image.

Sync channels If checked, the channel slider (c) is synchronized across the synchronized window set.

Image coordinates If checked, spatial calibration units will be used instead of pixels coordinates. For proper registration, this option should be unchecked when syncing images with different pixel sizes (see Image ⊿ Properties... [P]).

Sync z-slices If checked, the depth slider (z) is synchronized across the synchronized window set.
Analyze ⊃ Tools ⊃ Synchronize Windows, IJ 1.46j. This command can be used to measure ROIs across images, transfer ROIs from a reference image (e.g., a Maximum Intensity Projection, MIP) to multiple images or to use an un-zoomed copy of the active image as a navigator palette.

**Sync t-frames** If checked, the frame slider (t) is synchronized across the synchronized window set.

**Image scaling** If checked, positions to different windows are translated via offscreen coordinates, providing correct registration at different zoom levels.

**See also:** XII Transferring Selections Between Images, ROI Manager

### 31 Plugins ⊃

#### 31.1 Macros ⊃

This submenu contains commands for installing, running and recording macros, as well as any macro commands added by Plugins ⊃ Macros ⊃ Install... Macros contained in a file named STARTUPMACROS.txt, in the macros folder, are automatically added to this submenu when ImageJ starts up. By design, only one set of macros can be installed at a time. As such, the last set of macros installed by Plugins ⊃ Macros ⊃ Install... (or by the More Tools Menu) will always replace previously installed macros.

#### 31.1.1 Install...

Adds one or more macros contained in a file to the bottom of this submenu. To install a set of macros, and at the same time view their source code, open the macro file with File ⊃ Open and use Editor’s Macros ⊃ Install Macros command. Macros in the file ImageJ/macros/STARTUPMACROS.txt are automatically installed when ImageJ starts up. Similarly, with ImageJ 1.44f and later, newly opened macro sets with two or more macros are also automatically installed in this menu.

**See also:** More Tools Menu

#### 31.1.2 Run...

Loads and runs a macro without opening it in Editor. To run a macro, and at the same time view its source code, open it with File ⊃ Open and use the editor’s Macros ⊃ Run Macro command.

#### 31.1.3 Startup Macros...

Opens ImageJ/macros/STARTUPMACROS.txt. The same file can be opened by holding Shift while selecting Startup Macros from the drop-down menu (see More Tools Menu).

#### 31.1.4 Record...

Opens the ImageJ command recorder. To create a macro, open the recorder, use one or more ImageJ commands, then click Create. When the recorder is open, each menu command you use generates a macro run() function call. The run() function has one or two string arguments. The first is the command name. The optional second argument contains dialog box parameters. Examples:

- Create a rectangular, oval or line selection and the recorder will generate a makeRectangle(), makeOval() or makeLine() function call.
- Click on Auto or Set in the Image ⊃ Adjust ⊃ Threshold... [T] window to generate a setThreshold() call, and on Reset to generate a resetThreshold() call.
- Select an image from the Window menu to generate a selectWindow() call.
- Click in the Image ⊃ Color ⊃ Color Picker... [K] window to generate setForegroundColor() and setBackgroundColor() calls.
Note that you can interact with the recorder window by deleting or commenting lines of code or pasting text from Editor. This may be specially useful when writing your own macros or to generate simple ‘Session Logs’. In this case, you would start the Recorder and let ImageJ keep track of the performed actions by generating macro code.


31.2 Shortcuts

This submenu contains commands for creating keyboard shortcuts and for installing and removing plugins.

See also: Toolbar Shortcuts

31.2.1 List Shortcuts...

This command generates a table with the ImageJ keyboard shortcuts in one column and the commands they call in another. Commands prefixed by ‘*’ refer to shortcuts created with Create Shortcuts… while commands prefixed by ‘^’ refer to installed macros (listed in Plugins ⊳ Macros ⊳) and override ImageJ default hot-keys.

Note that unless Require control key for shortcuts in Edit ⊳ Options ⊳ Misc… is checked, you do not have to hold down Ctrl to use a keyboard shortcut. E.g., to open an image (File ⊳ Open… [o]) simply press O.

See also: Keyboard Shortcuts, Create Shortcuts…

31.2.2 Create Shortcuts...

Assigns a keyboard shortcut to an ImageJ menu command and lists the shortcut in the Shortcuts submenu.

Select the command from the popup menu and enter the shortcut in the text field. A shortcut can be a lower or uppercase letter or ‘F1’ through ‘F12’. Use Plugins ⊳ Utilities ⊳ List Shortcuts… to get a list of shortcuts that are already in use.

31.2.3 Install Plugin…

Installs a plugin in a user-specified submenu. Plugins with a showAbout() method are also automatically added to the Help ⊳ About Plugins ⊳ submenu.

Use the first popup menu to select the plugin and the second to select the submenu it is to be installed in. The command must be different from any existing ImageJ command. Shortcut (optional) must be a single letter or ‘F1’ through ‘F12’. Argument (optional) is the string that will passed to the plugin’s run method.

31.2.4 Remove…

Removes commands added to the Shortcuts submenu by Create Shortcuts… Also removes commands added by Install Plugin… and removes plugins installed in the Plugins menu. The menus are not updated until ImageJ is restarted.

31.3 Utilities

31.3.1 Control Panel… [U]

This command [47] opens a window containing ImageJ commands in a hierarchical tree structure. Click on a leaf node to launch the corresponding ImageJ command (or plugin). Double click on a tree branch node (folder) to expand or collapse it. Click and drag on a tree branch node (folder) to display its descendants in a separate (child) window. In a child window, click on ‘Show Parent’ to re-open the parent window.

See also: Find Commands… [l]

31.3.2 Find Commands… [l]

The quickest way to find a command without having to navigate through all the menus [48].

Evoke the prompt by pressing L (as in ‘command Launcher’, or ‘Locator’). If you type part of a command name, the list will only show commands that match that substring. If only a single command matches then that command can be run by pressing Enter. If multiple commands match, click with the mouse to select a command to run. Alternatively pressing the up or down keys will move keyboard focus to the list and the selected command can be run by pressing Enter. Pressing Backspace switches focus back to the prompt. Double clicking on a command will run that command. Pressing [Esc] closes the window.
Show full information  If checked, the Command Finder will display the location of the listed menu entries.

Fuzzy matching  (Fiji only) Activates approximate string matching. Useful if you are not sure about the command spelling.

Close when running  If checked, the Command Finder will dismiss after choosing Run or pressing Enter.

Export  Prints the filtered list of commands to an ImageJ table.

See also:  Control Panel... [U], Search..., Finding Commands

31.3.3  Search...

Searches for macros (.txt, .ijm), scripts (.js, .py, .rb, .clj, .bash), plugins source (.java) and .html files containing a particular string.

Search is performed recursively (subdirectories are included) and results displayed in the Log Window. In the Log window, double click on a file path to have it open.

Search contents  Specifies if the search should be restricted to filenames or extended to file contents. If checked, the line number where the string was found is displayed.

Ignore case  Specifies if the search should be case-insensitive.

Search macros folder  Extends the search scope to ImageJ/macros/.

Search plugins folder  Extends the search scope to ImageJ/plugins/.

Search scripts folder  Extends the search scope to ImageJ/scripts/ (if present).


Note that you can perform searches in other directories by choosing none of the folders above mentioned. In this case, you will be asked to choose a target directory on a second dialog prompt.

See also:  Search.txt, the macro in ij.jar implementing this command, Find Commands... [I], Finding Commands

31.3.4  Monitor Events...

By implementing the IJEventListener, CommandListener and ImageLister interfaces, this command is able to monitor foreground and background color changes, tool switches, Log window closings, command executions and image window openings, closings and updates.

See also:  Debug mode (Edit→Options→Misc...)

31.3.5  Monitor Memory...

Displays a continuously updated graph of ImageJ’s memory utilization, which can be useful for detecting memory leaks.

Memory usage and running threads are displayed above the graph. As for the IJ Status bar, clicking on the window will reclaim unused memory by running the Java garbage collector.

Ideally you should be able to open several images, process them, close them, and the amount of memory used will be the same as when you started.

See also:  Edit→Options→Memory & Threads...

31.3.6  Capture Screen [g]

Copies the screen to an RGB image and displays that image in a new window. Holding Ctrl will capture the screen while a modal dialog box is active if the dialog is based on ImageJ’s GenericDialog class.

See also:  Capture Image, Flatten [F]

31.3.7  Capture Image

Copies a WYSIWYG version of active image to an RGB image and displays that image a new window.

See also:  Flatten [F], Capture Screen [g]

31.3.8  ImageJ Properties...

This command displays various ImageJ properties (Java version, OS name and version, path separator, location of directories, screen size, etc.) in a text window. Holding Alt lists all Java properties.

See also:  Status bar, Help→About ImageJ...

31.3.9  Threads...

This command lists, in a text window, the currently running threads and their priorities.

31.3.10  Benchmark

Runs 62 image processing operations on the current image and displays the elapsed time in the ImageJ status bar. Additional benchmarks, test results, and source code are available in the Benchmarks package of plugins.

See also:  FAQs on the ImageJ Wikipage
31.3.11 Reset...

Use this command to unlock a locked Image, or to reclaim memory used by the clipboard and undo buffers (cf. Undo and Redo).

See also: File > Options > Reset...

31.4 New

This submenu contains commands opening editor windows that can be used to edit and run macros, scripts and plugins. It also has a command that opens a text window of a specified size and a command that opens a table that macros can write to. The editor windows opened by Plugin, Plugin Filter and Plugin Frame contain prototype Java code for the three types of plugins supported by ImageJ.

See also: Editor, Extending ImageJ

31.4.1 Macro

Opens a blank editor window with the title ‘Macro.txt’.

See also: Macros, Editor, Text Window..., File > New > Text Window [N]

31.4.2 Macro Tool

Opens [Macro_Tool.txt], an example macro tool that creates circular selections.

See also: Macro Tools Documentation

31.4.3 JavaScript

Opens a blank editor window with the title ‘Script.js’.

See also: Scripts, Editor

31.4.4 Plugin

Opens an editor window containing a prototype plugin that implements the PlugIn interface. Plugins of this type open, capture or generate images. The prototype displays Hello world!.

Press [Ctrl] + [R] (File > Run Macro) to compile and run it. Note that the name you choose for the plugin should include at least one underscore. Another example is the Step Maker plugin.

31.4.5 Plugin Filter

Opens an editor window containing a prototype plugin that implements the PlugInFilter interface. Plugins of this type process the active image. The prototype inverts the active image twice. Another example is the Image Inverter.

31.4.6 Plugin Frame

Opens an editor window containing a prototype plugin that extends the PluginFrame class. Plugins of this type displays a window containing controls such as buttons and sliders. The prototype opens a window containing a text area. Another example is the IP Demo plugin.

31.4.7 Plugin Tool

Opens a prototype plugin tool, demonstrating ij.plugin.tool.PlugInTool [49]. A plugin tool is a Java plugin that installs in the ImageJ toolbar to interact with the image canvas (see Custom Tools). Plugin tools with names ending in Tool are listed on the More Tools Menu if placed in the ImageJ/plugins/Tools/ directory.

See also: Plugin Tools

31.4.8 Text Window...

Opens a text window of a specified size that macros can write to. PrintToTextWindow, Clock and ProgressBar are examples of macros that write to a text window.

See also: Plugins > New > Macro, File > New > Text Window [N]

31.4.9 Table...

Opens a blank table, similar to the Results table, that macros can write to. SineCosineTable2 is an example of such a macro.

See also: Plugins > New > Text Window...

31.5 Compile and Run...

Compiles and runs a plugin. Runs a plugin if the name of the selected file ends in .class. Requires that ImageJ be running on a Java Virtual Machine that includes the javac compiler, which is contained in the Tools.jar archive. Tools.jar is included with the Windows and Linux versions of ImageJ bundled with Java, and it is pre-installed on Mac OS X. Troubleshooting information can be found under ‘Compiling Plugins’ in the Linux and Windows release notes.

The Edit > Options > Compiler... command can be used to configure the javac compiler. Since ImageJ 1.44c and later, Compile and Run... adds the Bio-Formats plugin (loci_tools.jar) to the Java compiler’s classpath.
32  **Window**

This menu contains four commands plus a list of all open windows. The currently active image
will have a checkmark next to its name. To activate a window, pull down this menu and select
the window by name.

32.1  **Show All [ ]**

Makes all the windows associated with ImageJ visible.

32.2  **Put Behind [tab]**

Displays the next open image. Repeatedly press the tab key to cycle through all open images.
Note that pressing [Enter] on any image will bring the Main ImageJ window to the foreground.

**See also:** I Frontmost Window and Window Activation

32.3  **Cascade**

Moves all open images to the left side of the screen, slightly offset from each other, and displayed
in the order they are listed at the bottom of this menu.

32.4  **Tile**

Shrinks all open image windows and repositions them to fit on the screen without overlapping.

33  **Help**

Many of the commands in this menu use ImageJ’s **BrowserLauncher** to open a Web page using
the user’s default browser. On Linux, **BrowserLauncher** looks for ‘netscape’, ‘firefox’, ‘konqueror’,
‘mozilla’, ‘opera’, ‘epiphany’ or ‘lynx’ and uses the first one it finds.

33.1  **ImageJ Website...**

Opens the ImageJ home page.

33.2  **ImageJ News...**

Opens the News section of the ImageJ website.

33.3  **Documentation...**

Opens the Documentation section of the ImageJ website.

33.4  **Installation...**

Opens the Installation section of the ImageJ website specifically dedicated to the OS in which
ImageJ is running, i.e either Linux, Mac OS 9, Mac OS X or Windows. Worth reading are the
sections Known Problems and Troubleshooting.

33.5  **Mailing List...**

Opens the ImageJ Interest Group page on the NIH LISTSERV facility. Here you can search the
mailing list archives, post a message to the list, join or leave the list, or update options.

33.6  **Dev. Resources...**

Opens the Developer Resources section of the ImageJ website.

33.7  **Plugins...**

Opens the Plugins page on the ImageJ website, which lists more than 500 ImageJ plugins.

33.8  **Macros...**

Opens the macros directory on the ImageJ website, which contains more than 400 ImageJ macros.

33.9  **Macro Functions...**

Opens the Macro Functions reference page, an indispensable guide to the built in functions that
can be called from the ImageJ macro language.

**See also:** Macros, Function Finder... [F]
33.10 Update ImageJ...


See also: Installing and Maintaining ImageJ

33.11 Refresh Menus

Use this command to update ImageJ’s menus after adding (or removing) plugins or macros to the plugins folder. Prior to ImageJ 1.44b this command was named ‘Update Menus’.

33.12 About Plugins>

This submenu displays information about some of the plugins in the ImageJ plugins folder. To be included in this submenu, a plugin must be packaged as a JAR file. There is an example at rsb.info.nih.gov/ij/plugins/jar-demo.html.

33.13 About ImageJ...

Opens an image containing information about the ImageJ version, the author, the website, Java version and memory available. Note that clicking in the Status bar is a quicker way to show this information.

See also: Plugins > Utilities > ImageJ Properties...

Part VI

Keyboard Shortcuts

The following table summarizes the keyboard shortcuts built into ImageJ. You can create additional shortcuts, or override built-in ones, by creating simple macros and adding them to the StartupMacros.txt. You can also assign a function key to a menu command using Plugins > Shortcuts > Create Shortcuts…

Several of these shortcuts accept key modifiers as described in Key Modifiers. Also note that, except when using the Text Tool, you do not need to hold down the control key to use a keyboard shortcut, unless Require control key for shortcuts in Edit > Options > Misc… is checked.

See also: Using Keyboard Shortcuts, Finding Commands, KeyboardShortcuts.txt macro (demonstrating how assign shortcuts to custom macros), Toolbar Shortcuts

List of ImageJ 1.46r built-in shortcuts. This table can be obtained within ImageJ using the Plugins > Shortcuts > List Shortcuts… command.

<table>
<thead>
<tr>
<th>Command / Operation</th>
<th>Shortcut</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File &gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>New &gt; Image... [n]</td>
<td>N</td>
<td>Create new image or stack</td>
</tr>
<tr>
<td>New &gt; Text Window [N]</td>
<td>Shift N</td>
<td>Create new text window</td>
</tr>
<tr>
<td>New &gt; System Clipboard [V]</td>
<td>Shift V</td>
<td>Create image from system clipboard</td>
</tr>
<tr>
<td>Open... [o]</td>
<td>O</td>
<td>Open file (any format recognized by ImageJ)</td>
</tr>
<tr>
<td>Open Next [O]</td>
<td>Shift O</td>
<td>Open next image in folder</td>
</tr>
<tr>
<td>Open Samples &gt; Blobs (25K)</td>
<td>Shift B</td>
<td>Opens the Blobs.gif example image</td>
</tr>
<tr>
<td>Close [w]</td>
<td>W</td>
<td>Close the active window</td>
</tr>
<tr>
<td>Save [s]</td>
<td>S</td>
<td>Save active image in Tiff format</td>
</tr>
<tr>
<td>Revert [r]</td>
<td>R</td>
<td>Revert to saved version of image</td>
</tr>
<tr>
<td>Print... [p]</td>
<td>P</td>
<td>Print active image</td>
</tr>
<tr>
<td>Edit &gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Undo [z]</td>
<td>Z</td>
<td>Undo last operation</td>
</tr>
<tr>
<td>Cut [x]</td>
<td>X</td>
<td>Copy selection to internal clipboard and clear</td>
</tr>
<tr>
<td>Copy [c]</td>
<td>C</td>
<td>Copy selection to internal clipboard</td>
</tr>
<tr>
<td>Paste [v]</td>
<td>V</td>
<td>Paste contents of internal clipboard</td>
</tr>
<tr>
<td>Clear</td>
<td>Backspace</td>
<td>Erase selection to background color</td>
</tr>
<tr>
<td>Fill [f]</td>
<td>F</td>
<td>Fill selection in foreground color</td>
</tr>
<tr>
<td>Draw [d]</td>
<td>D</td>
<td>Draw selection</td>
</tr>
<tr>
<td>Invert [i]</td>
<td>Shift I</td>
<td>Invert image or selection</td>
</tr>
<tr>
<td>Selection &gt; Select All [a]</td>
<td>A</td>
<td>Select entire image</td>
</tr>
<tr>
<td>Selection &gt; Select None [A]</td>
<td>Shift A</td>
<td>Remove selection</td>
</tr>
<tr>
<td>Selection &gt; Restore Selection [E]</td>
<td>Shift E</td>
<td>Restore previous selection</td>
</tr>
<tr>
<td>Selection &gt; Properties... [y]</td>
<td>Y</td>
<td>Defines selection properties</td>
</tr>
<tr>
<td>Selection &gt; Add to Manager [t]</td>
<td>T</td>
<td>Add selection to ROI Manager</td>
</tr>
<tr>
<td>Image &gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adjust &gt; Brightness/Contrast... [C]</td>
<td>Shift C</td>
<td>Adjust brightness and contrast</td>
</tr>
<tr>
<td>Adjust &gt; Threshold... [T]</td>
<td>Shift T</td>
<td>Adjust threshold levels</td>
</tr>
<tr>
<td>Show Info... [I]</td>
<td>I</td>
<td>Display information about active image</td>
</tr>
<tr>
<td>Properties... [P]</td>
<td>Shift P</td>
<td>Display image properties</td>
</tr>
</tbody>
</table>
List of ImageJ 1.46r built-in shortcuts. This table can be obtained within ImageJ using the Plugins>Shortcuts> List Shortcuts... command.

<table>
<thead>
<tr>
<th>Command / Operation</th>
<th>Shortcut</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color&gt; Color Picker... [K]</td>
<td>Shift K</td>
<td>Open Color Picker</td>
</tr>
<tr>
<td>Stacks&gt; Next Slice [&gt;]</td>
<td>&gt;</td>
<td>Go to next stack slice</td>
</tr>
<tr>
<td>Stacks&gt; Previous Slice [&lt;]</td>
<td>&lt;</td>
<td>Go to previous stack slice</td>
</tr>
<tr>
<td>Stacks&gt; Reslice... []</td>
<td>/</td>
<td>Reslice stack</td>
</tr>
<tr>
<td>Stacks&gt; Orthogonal Views [H]</td>
<td>Shift H</td>
<td>Toggle orthogonal view display</td>
</tr>
<tr>
<td>Stacks&gt; Channels Tool... [1]</td>
<td>1</td>
<td>Start/stop stack animation</td>
</tr>
<tr>
<td>Hyperstacks&gt; Channels Tool... [2]</td>
<td>Shift Z</td>
<td>Open the 'Channels' tool</td>
</tr>
<tr>
<td>Hyperstacks&gt; Next hyperstack channel</td>
<td>&gt; or &gt;</td>
<td>Next hyperstack channel</td>
</tr>
<tr>
<td>Hyperstacks&gt; Previous hyperstack channel</td>
<td>&lt; or &lt;</td>
<td>Previous hyperstack channel</td>
</tr>
<tr>
<td>Hyperstacks&gt;</td>
<td>Ctrl &gt;</td>
<td>Next hyperstack slice</td>
</tr>
<tr>
<td>Hyperstacks&gt;</td>
<td>Ctrl &lt;</td>
<td>Previous hyperstack slice</td>
</tr>
<tr>
<td>Hyperstacks&gt;</td>
<td>Add &gt;</td>
<td>Next hyperstack frame</td>
</tr>
<tr>
<td>Hyperstacks&gt;</td>
<td>Add &lt;</td>
<td>Previous hyperstack frame</td>
</tr>
<tr>
<td>Crop [X]</td>
<td>Shift X</td>
<td>Crop active image or selection</td>
</tr>
<tr>
<td>Duplicate... [D]</td>
<td>Shift D</td>
<td>Duplicate active image or selection</td>
</tr>
<tr>
<td>Scale... [E]</td>
<td>E</td>
<td>Scale image or selection</td>
</tr>
<tr>
<td>Zoom&gt; In [+ ]</td>
<td>+</td>
<td>Make image larger</td>
</tr>
<tr>
<td>Zoom&gt; Out [- ]</td>
<td>-</td>
<td>Make image smaller</td>
</tr>
<tr>
<td>Zoom&gt; View 100% [S]</td>
<td>0</td>
<td>Zoom to 1:1</td>
</tr>
<tr>
<td>Overlay&gt; Add Selection... [b]</td>
<td>B</td>
<td>Adds active selection to image overlay</td>
</tr>
<tr>
<td>Process&gt;&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smooth [S]</td>
<td>Shift S</td>
<td>3x3 unweighted smoothing</td>
</tr>
<tr>
<td>Repeat Command [R]</td>
<td>Shift R</td>
<td>Repeat previous command</td>
</tr>
</tbody>
</table>

**Analyze>>**
| Measure... [m] | M | Display statistics of active image/selection |
| Histogram [h] | H | Display histogram of active image/selection |
| Plot Profile [k] | K | Display density profile plot of active selection |
| Gels> Select First Lane | 1 | Select first gel lane |
| Gels> Select Next Lane | 2 | Select next gel lane |

**Plugins>>**
| Control Panel... [U] | Shift U | Open Control Panel |
| Capture Screen [a] | Shift G | Grab screenshot (with [Ctrl] if a dialog box is active) |
| Find Commands... [l] | L | List, find and launch commands |

**Window>>**
| Show All... [ ] | 1 | Make all windows visible |
| Put Behind [tab] | Tab | Switch to next image window |
| Main ImageJ window | Enter | Bring ImageJ window to front |

### 34 Key Modifiers

#### 34.1 Alt Key Modifications

**File>>**
| Open Next [O] | Open previous |
| Revert [r] | Skip dialog prompt |

**Edit>>**
| Copy [c] | Copy to system clipboard |

**Image>>**
| Color> Split Channels | Keep original image |

**Image>>**
| Stacks> Add Slice | Insert before current slice |
| Next Slice [>] | Skip ten slices |
| Previous Slice [<] | Skip ten slices |
| Start Animation [1] | Show options dialog |
| Duplicate... [D] | Skip dialog prompt |
| Add Selection... [b] | Show options dialog |

**Process>>**
| Enhance Contrast... | Do classic histogram equalization |

**Analyze>>**
| Histogram [h] | Show dialog prompt |

**Plugins>>**
| Plot Profile [k] | For rectangular selections, generate row average plot. For wide straight lines, display rotated contents |

**Utilities>>**
| ImageJ Properties... | List all Java properties |

Area Selection Tools Subtract current selection from the previous one
Segmented Line Selection Tool and PolyLine Selection Tool Current aspect ratio is maintained while resizing

Straight Line Selection Tool Keeps the line length fixed while moving either end of the line. Forces the two points that define the line to have integer coordinate values when creating a line on a zoomed image
Segmented Line Selection Tool and PolyLine Selection Tool Assume lanes are horizontal

Point Tool Alt-clicking on a node deletes it
Color Picker Tool Alt-clicking on an image 'picks-up' background color
All Tools Show location and size in pixels rather than calibrated units

### 34.2 Shift Key Modifications

**Image>>**
| Adjust> Min [T] | Adjusting Min also adjusts Max |
| Adjust> Brightness/Contrast... [C] | Apply adjustments to all channels of a composite image |

**Installed**
| Macros and Scripts | Open instead of run |

**Rectangular Selection Tool and Oval Selection Tool**
| 1:1 aspect ratio |

**Area Selection Tools**
| Add selection to previous one |

**Point Tool**
| Shift-clicking on a node duplicates it
| Forces line to be horizontal or vertical |

**Segmented Line Selection Tool**
| Shift-clicking on a node duplicates it
| Point Tool Shift-clicking adds points (Multi-point Tool behavior) |

**Magnifying Glass**
| Shift-clicking and dragging runs Image> Zoom> To Selection |

**SEE ALSO:** Manipulating ROIs, Tools
34.3 Ctrl (or Cmd) Key Modifications

Rectangular Selection Tool and Oval Selection Tool Selection is resized around its center
Straight Line Selection Tool Line is rotated/resized around its center

SEE ALSO: Manipulating ROIs, Tools

34.4 Space Bar

Any Tool Switch to the Scrolling Tool

34.5 Arrow Keys

Moving Selections The four arrow keys move selection outlines one pixel at a time
Resizing Selections Rectangular and oval selections are resized by holding \( \text{Alt} \) while using the arrow keys
Stacks Navigation The \( \text{←} \) and \( \text{→} \) keys substitute for \( \text{≤} \) and \( \geq \) for moving through a stack. If there is a selection, you must also hold \( \text{Shift} \)
Hyperstacks Navigation The \( \text{←} \) and \( \text{→} \) keys change the channel. Hold \( \text{Ctrl} \) to move through the slices and \( \text{Alt} \) to move through the frames
Zooming The \( \text{↑} \) and \( \text{↓} \) keys zoom the image in and out. If there is a selection, you must also hold either \( \text{Shift} \) or \( \text{Ctrl} \)

SEE ALSO: Manipulating ROIs, Zoom>

35 Toolbar Shortcuts

Keyboard shortcuts cannot be used directly to activate tools in the ImageJ Toolbar (with the exception of the Magnifying Glass and the Scrolling Tool). However, shortcuts can be assigned to macros that use the \text{setTool()} macro function.

The set of macros listed below (taken from ImageJ/macros/StartupMacros.txt) exemplify how to assign the function keys \( \text{F1} \) through \( \text{F2} \) to some of the most commonly used Tools. Once copied to the \text{ImageJ/macros/StartupMacros.txt} file, they will be automatically installed at startup.

This approach, however, requires the user to memorize a large number of shortcuts. In addition, it may be difficult to assign so many hot-keys without conflicting with previously defined ones (see Plugins>Shortcuts). An alternative way to control the toolbar using the keyboard is to create macros that progressively activate tools from a predefined sequence. The next example demonstrates such strategy. It is composed of two macros activated by \( \text{F1} \) and \( \text{F2} \) that iterate through the toolbar from left to right (forward cycle) and right to left (reverse cycle).

A tool can be defined either by its name or by its position in the toolbar using \text{setTool(id)}, which allows assigning keyboard shortcuts to Custom Tools and items loaded by the More Tools Menu (e.g., \text{setTool(21)}), activates whatever tool has been installed on the last slot of the toolbar. It is also possible to temporarily activate a tool. The macro below (taken from the ImageJ/macros/StartupMacros.txt) file, they will be automatically installed at startup.

<table>
<thead>
<tr>
<th>10</th>
<th>Cycling Through the Toolbar Using Keyboard Shortcuts</th>
</tr>
</thead>
<tbody>
<tr>
<td>/* These two macros loop through the tools listed in an array using “F1” ← and “F2” as keyboard shortcuts (forward and reverse cycling). */</td>
<td></td>
</tr>
<tr>
<td>var index;</td>
<td></td>
</tr>
<tr>
<td>macro &quot;Cycle Tools Fwd [F1]&quot; {</td>
<td></td>
</tr>
<tr>
<td>setTool(tools[index++]);</td>
<td></td>
</tr>
<tr>
<td>if (index==tools.length) index = 0;</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
<tr>
<td>macro &quot;Cycle Tools Rev [F2]&quot; {</td>
<td></td>
</tr>
<tr>
<td>if (index&lt;0) index = tools.length-1;</td>
<td></td>
</tr>
<tr>
<td>setTool(tools[index--]);</td>
<td></td>
</tr>
</tbody>
</table>
| }

(11) Temporary Activation of a Tool

A tool can be defined either by its name or by its position in the toolbar using \text{setTool(id)}, which allows assigning keyboard shortcuts to Custom Tools and items loaded by the More Tools Menu (e.g., \text{setTool(21)}), activates whatever tool has been installed on the last slot of the toolbar. It is also possible to temporarily activate a tool. The macro below (taken from the ImageJ/macros/StartupMacros.txt) file, they will be automatically installed at startup.

A tool can be defined either by its name or by its position in the toolbar using \text{setTool(id)}, which allows assigning keyboard shortcuts to Custom Tools and items loaded by the More Tools Menu (e.g., \text{setTool(21)}), activates whatever tool has been installed on the last slot of the toolbar. It is also possible to temporarily activate a tool. The macro below (taken from the ImageJ/macros/StartupMacros.txt) file, they will be automatically installed at startup.
Credits

[C1] The ImageJ installer for Windows is created using the Inno Setup installer generator. The ImageJ.exe the Windows that launches ImageJ ij.jar was contributed by George Silva.

[C2] Support for ZIP-compressed TIFFs was contributed by Jason Newton inIJ 1.45g.

[C3] The macro editor’s Function Finder (Macros.Find Functions...) was written by Jérôme Mutterer.

[C4] The Elliptical Selection Tool was contributed by Norbert Vischer.

[C5] The Brush Selection Tool is based on the ROI Brush Tool plugin from Tom Larkworthy and Johannes Schindelin.

[C6] Jean-Yves Tinevez and Johannes Schindelin (authors of the Fiji Arrow Tool) contributed code to the Arrow Tool.

[C7] Michael Schmid, added 4-connected and 8-connected tracing with tolerance to the Wand Tool.

[C8] Macro Toolsets distributed with ImageJ have been contributed by Gilles Carpentier, Jérôme Mutterer and Tiago Ferreira.

[C9] The Pixel Inspector is a plugin tool conversion of Michael Schmid’s Pixel Inspector plugin.

[C10] InIJ 1.43f and earlier, the File>Import>Results... command was based on Jérôme Mutterer’s Import_Results_Table macro.

[C11] Marcel van Herk added URLs support to the File>Import>Stack From List... command inIJ 1.45f.


[C13] Karen Collins contributed improvements to the FITS_Writer (File>Save As FITS... command).

[C14] The Edit>Selection>Fit Circle command, based on a MATLAB script by Nikolai Chernov, was contributed by Michael Doube and Ved Sharma.

[C15] The Edit>Selection>Create Selection command is based on the Threshold_To_Selection plugin written by Johannes Schindelin.

[C16]IJ 1.46f adopted Johannes Schindelin’s RMI-based OtherInstance class from Fiji, which works on multi-user machines and is more secure.

[C17] The Color Picker (Image>Color>Color Picker...) was written by Gali Baler, a 2003–2004 intern from Bethesda-Chevy Chase High School.

[C18] The 16-different thresholding methods available in the Image>Adjust Threshold... command were implemented by Gabriel Landini.

[C19] Michael Schmid contributed improvements to the downsizing kernel used by Image>Adjust Size... and Image>Scale... as well as undo support for Image>Scale... [E].

[C20] The Image>Adjust>Color Threshold... command implements Gabriel Landini’s Threshold Colour plugin.

[C21] The Reslice and the ZProject plugin (Image>Stacks>Reslice... and Z Project... commands) were contributed by Patrick Kelly and Harvey Karten of the University of California, San Diego.

[C22] The Image>Stacks>Orthogonal Views [H] command is based on Dimiter Prodanov’s StackSlicer plugin and Albert Cardona’s Updater class. Michael Doube added support for XZ and YZ view control as well as mouse wheel control.

[C23] The Image>Stacks>3D Project... was written by Michael Castle and Janice Keller of the University of Michigan Mental Health Research Institute (MHRI). Bill Mohler added support for hyperstacks and 16/32-bit images inIJ 1.44m.

[C24] The Image>Stacks>Tools>Concatenate... command implemented inIJ 1.46e is based on the Concatenate plugin by Jonathan Jackson.

[C25] The Image>Stacks>Tools>Makem Substack... command is based on the Substack Maker plugin by Anthony Padua, Daniel Barbioriak and Ved Sharma.

[C26] The Image>Hyperstacks>Reduce Dimensionality... command is based Jérôme Mutterer’s Reduce HyperStack macro.

[C27] The Image>Transform>Bin... command is based on Nico Stuurman’s Binner plugin.

[C28] The Image>Zoom>Set... command is based on Albert Cardona’s Zoom Exact plugin.

[C29] The Process>Find Maxima... command is based on a plugin contributed by Michael Schmid.

[C30] The equalization code implemented inProcess>Enhance Contrast... was contributed by Richard Kirk.

[C31] The Process>Noise>Remove NaNs... was contributed by Michael Schmid.

[C32] The Process>Math>Macro... command is modeled after UI Dittner’s Expression plugin.

[C33] The Process>FFT>Bandpass Filter... is a built-in version of Joachim Walter’s FFT Filter plugin.

[C34] The Process>Binary>Fill Holes algorithm was contributed by Gabriel Landini.


[C36] Multi-threading support for allProcess>Filters... commands was contributed by Stephan Saalfeld and Michael Schmid inImageJ 1.45c.

[C37] The faster and more accurate version ofProcess>Filters>Gaussian Blur... implemented inImageJ 1.38r was contributed by Michael Schmid.

[C38] The NonBlockingGenericDialog class used by the Process>Batch>Macro... command was added by Johannes Schindelin.

[C39] The rolling ball code ofProcess>Subtract Background... is based on the NIH Image Pascal version by Michael Castle and Janice Keller. The sliding paraboloid algorithm was written by Michael Schmid.

[C40] The Analyze>Distribution... command was written by Gabriel Landini.

[C41] The scaled color bar implemented in Analyze>Histogram [h] was contributed by Bob Dougherty.

[C42] The much improved CurveFitter (Analyse>Tools>Curve Fitting...) implemented inIJ 1.46f was contributed by Michael Schmid. The Rodbard and Gaussian functions were originally contributed by David Rodbard (NIH) and Stefan Wörz (DKFZ), respectively.

[C43] The ROI Manager<ROI command (Analyse>Tools>ROI Manager...) was added by Johannes Schindelin.

[C44] The ROI Manager<Multi Measure command (Analyse>Tools>ROI Manager...) is based on Bob Dougherty’s Multi_Measure plugin.

[C45] The ROI Manager<Multi Plot command (Analyse>Tools>ROI Manager...) was contributed by Philippe Gendre.

[C46] The Analyze>Tools>Synchronize Windows command (an improved version of the SyncWindows... plugin by Patrick Kelly) was contributed by Joachim Walter.

[C47] The Control Panel<Plugins>Utilities>Control Panel... [U] was written by Cesar M. Tigare.

[C48] The Command Finder<Plugins>Utilities>Find Command... [F] was written by Mark Longair.

[C49] The PlugInTools class was inspired by Johannes Schindelin’s AbstractTool class in Fiji.

[C50] Other additions, improvements and reproducible bug reports have been contributed by Adrian Daerr, Airen Peraza, Ajay Gopal, Albert Cardona, Alberto Duina, Alden Dimas, Andreas Maijer, Andrew French, Andrii Savchenko, Arttu Miettinen, Aryeh Weiss, Balazs Nyiri, Barry DeZonia, Bill Mohler, Bob Hamilton, Bob Loushin, Bruno Vellutini, Burri Olivier, Carlos Becker, Carne Draug, Charles Anderson, Cheryl McCreary, Christian Moll, Christophe Leterrier, Christopher Harrison, Damon Podurlo, Daniel Horning, Daniel Kalthoff, Daniel Senff, David Gauntt, David McDonald, Denzy Hugg, Dimiter Prodanov, Divokar Ramachandran, Dorian Iyer, Duncan Mak,
Eik Schumann, Emmanuel Levy, Erik Meijering, Fabian Svara, Francis Burton, Frank Sprenger, 
Franklin Shaffer, Frederic Hessman, Gabriel Landini, Gilles Carpentier, Gregory Rennef, Hiroki 
Hakoshima, Ian Lim, Ingo Bartholomaeus, Jakob Preus, Jamie Robinson, Jan Eglinger, Jan 
Funke, Jarek Sacha, Jay Urrah, Jean-Pierre Clamme, Jerome Mutteer, Jesper Pedersen, Jim 
Passmore, Joachim Wenzer, Johannes Hermen, Johannes Schindelin, Johannes Weissmann, John 
Oropoulois, John Pearl, Jonathan Silver, Jose Wojnakzi, Juan Grande, Julian Cooper, Kai Uwe 
Barthel, Karen Collins, Kees Straatman, Kevin Moll, Kris Sweerts, Mark Krebs, Mark Langair, 
Martin Dressler, Mat Al-Tamimi, Matthew Smith, Michael Camner, Michael Doube, Michael 
Ellis, Michael Schmid, Michel Julier, Naganaanda Gurudev, Nico Stuurman, Norbert Vischer, 
Olaf Freyer, Oliver Bannach, Olivier Bardot, Paul Jurczak, Peter Haub, Rainer Engel, Raymond 
Coory, Reinhard Mayr, Richard Cole, Richie Mort, Robert Dougherty, Shannon Stewman, Simon 
Roussel, Stefan Starke, Stephan Saalfeld, Steven Green, Thomas Bodmer, Thorsten Wagner, 
Tiago Ferreira, Tomas Karlsson, Tseung Qimgong, Ulf Dittmer, Uwe Walschus, Valerio Musi, 

ImageJ Related Publications

The following references are a small sample (particularly biased towards the life sciences) of 
the bibliography directly related to ImageJ, the standard in scientific image analysis. These 
publications include: 1) technical articles and books describing routines implemented in ImageJ, 
2) research articles that have made extensive use of ImageJ as a scientific tool or 3) reviews that 
discuss ImageJ pertinently. A similar list is maintained on the Fiji website.

To cite ImageJ one of the citations is possible (see FAQs):

1. Rasband WS. ImageJ, U.S. National Institutes of Health, Bethesda, Maryland, USA, imagej.nih.gov/ij/, 

2. Abramoff MD, Magalhaes PJ and Ram SJ. “Image Processing with ImageJ,” Biophotonics International, 
11(7):36–42, 2004 (PDF) [1].

Methods, pp. 671, 2012 [102]

To cite Fiji:

pp. 676–82, 2012 [100]

To cite this document:


of Media Technology and Design Hagenberg, Austria, http://www.gm.fh-koeln.de/~konen/WPF-BV/


development using membrane immobilization and automatic image analysis. J Ind Microbiol Biotechnol, 


freeeware tool for digital reconstruction of neuronal arborizations from image stacks. Neuroinformatics, 

[10] Enem Bult and Binyamin Sahin. A new method of assessing the size of mandibular cysts on orthopan-
0b013e3181bd302z.


List of Abbreviations and Acronyms

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alt</td>
<td>Alt, Option or Meta key;</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface;</td>
</tr>
<tr>
<td>ASCII</td>
<td>American Standard Code for Information Interchange;</td>
</tr>
<tr>
<td>bit</td>
<td>Binary digit;</td>
</tr>
<tr>
<td>BMP</td>
<td>Bitmap Image File (Device Independent Bitmap, DIB);</td>
</tr>
<tr>
<td>bpp</td>
<td>Bits per pixel;</td>
</tr>
<tr>
<td>CCD</td>
<td>Charge-Coupled Device;</td>
</tr>
<tr>
<td>CP</td>
<td>Color Picker;</td>
</tr>
<tr>
<td>CSV</td>
<td>Comma-Separated Values;</td>
</tr>
<tr>
<td>CT</td>
<td>Computed Tomography;</td>
</tr>
<tr>
<td>Ctrl</td>
<td>Control key. In this guide also the Command key in Apple keyboards;</td>
</tr>
<tr>
<td>DICOM</td>
<td>Digital Imaging and Communications in Medicine;</td>
</tr>
<tr>
<td>DPI</td>
<td>Dots Per Inch;</td>
</tr>
<tr>
<td>EDM</td>
<td>Euclidian Distance Map;</td>
</tr>
<tr>
<td>FAQ</td>
<td>Frequently Asked Questions;</td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier Transform;</td>
</tr>
<tr>
<td>FHT</td>
<td>Fast Hartley Transform;</td>
</tr>
<tr>
<td>FIJI</td>
<td>Fiji Is Just Image;</td>
</tr>
<tr>
<td>FITS</td>
<td>Flexible Image Transport System;</td>
</tr>
<tr>
<td>fps</td>
<td>Frames Per Second;</td>
</tr>
<tr>
<td>GIF</td>
<td>Graphics Interchange Format;</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface;</td>
</tr>
<tr>
<td>HDR</td>
<td>High Dynamic Range;</td>
</tr>
<tr>
<td>HEX</td>
<td>Hexadecimal;</td>
</tr>
<tr>
<td>HIPR</td>
<td>Hypermedia Image Processing Reference;</td>
</tr>
<tr>
<td>HSB</td>
<td>Hue Saturation Brightness;</td>
</tr>
<tr>
<td>HTML</td>
<td>HyperText Markup Language;</td>
</tr>
<tr>
<td>IDE</td>
<td>Integrated Development Environment;</td>
</tr>
<tr>
<td>LJ</td>
<td>ImageJ;</td>
</tr>
<tr>
<td>ITK</td>
<td>Insight Segmentation and Registration;</td>
</tr>
<tr>
<td>JAR</td>
<td>Java Afflicive;</td>
</tr>
<tr>
<td>JPEG</td>
<td>Joint Photographic Experts Group;</td>
</tr>
<tr>
<td>Knime</td>
<td>Konstanz Information Miner;</td>
</tr>
<tr>
<td>KNIP</td>
<td>Knime Image Processing;</td>
</tr>
<tr>
<td>LUT</td>
<td>Lookuptable;</td>
</tr>
<tr>
<td>LZW</td>
<td>Lempel-Ziv-Welch;</td>
</tr>
<tr>
<td>MIP</td>
<td>Maximum Intensity Projection;</td>
</tr>
<tr>
<td>MJPEG</td>
<td>Motion-JPEG;</td>
</tr>
<tr>
<td>NaN</td>
<td>Not a Number;</td>
</tr>
<tr>
<td>OME</td>
<td>Open Microscopy Environment;</td>
</tr>
<tr>
<td>OS</td>
<td>Operating System;</td>
</tr>
<tr>
<td>p/c</td>
<td>Pixels per cycle;</td>
</tr>
<tr>
<td>PBM</td>
<td>Portable BitMap;</td>
</tr>
<tr>
<td>PGM</td>
<td>Portable GrayMap;</td>
</tr>
<tr>
<td>pixel</td>
<td>Picture element;</td>
</tr>
<tr>
<td>PNG</td>
<td>Portable Network Graphics;</td>
</tr>
<tr>
<td>ppi</td>
<td>Pixels per inch;</td>
</tr>
<tr>
<td>PPM</td>
<td>Portable Pixmap;</td>
</tr>
<tr>
<td>pt</td>
<td>Point;</td>
</tr>
<tr>
<td>RAM</td>
<td>Random-Access Memory;</td>
</tr>
<tr>
<td>regex</td>
<td>Regular expression;</td>
</tr>
<tr>
<td>RGB</td>
<td>Red Green Blue;</td>
</tr>
<tr>
<td>ROI</td>
<td>Region Of Interest;</td>
</tr>
<tr>
<td>Shift</td>
<td>Shift key;</td>
</tr>
<tr>
<td>TIFF</td>
<td>Tagged Image File Format;</td>
</tr>
<tr>
<td>UEPs</td>
<td>Ultimate Eroded Points;</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator;</td>
</tr>
<tr>
<td>voxel</td>
<td>Volumetric pixel;</td>
</tr>
<tr>
<td>WYSIWYG</td>
<td>What You See Is What You Get;</td>
</tr>
<tr>
<td>3D Viewer</td>
<td>89</td>
</tr>
<tr>
<td>3D Filters</td>
<td>20, 124</td>
</tr>
<tr>
<td>3D Gaussian blur</td>
<td>124</td>
</tr>
<tr>
<td>3D Object Counter</td>
<td>20</td>
</tr>
<tr>
<td>3D Projection</td>
<td>89</td>
</tr>
<tr>
<td>3D ROIs, 20</td>
<td>3D Viewer, 20, 89</td>
</tr>
<tr>
<td>Acceptable manipulation, 5</td>
<td></td>
</tr>
<tr>
<td>ActionBar, 25</td>
<td>Add see Math, 117</td>
</tr>
<tr>
<td>Add images see Image calculator, 127</td>
<td></td>
</tr>
<tr>
<td>Algorithm</td>
<td>Gift wrap, 63</td>
</tr>
<tr>
<td>Heckbert quantization, 14, 74</td>
<td></td>
</tr>
<tr>
<td>Newtonian-based Pratt fit, 63</td>
<td></td>
</tr>
<tr>
<td>Alpha blending value see Blend, 102</td>
<td></td>
</tr>
<tr>
<td>Analyze, 131</td>
<td></td>
</tr>
<tr>
<td>AND see Logical operations, 128, 150</td>
<td></td>
</tr>
<tr>
<td>Angle tool, 33</td>
<td></td>
</tr>
<tr>
<td>Animation, 55, 94</td>
<td></td>
</tr>
<tr>
<td>Annotation</td>
<td>Layers see ImageROI, 80</td>
</tr>
<tr>
<td>Shift</td>
<td>Shift key;</td>
</tr>
<tr>
<td>TIFF</td>
<td>Tagged Image File Format;</td>
</tr>
<tr>
<td>UEPs</td>
<td>Ultimate Eroded Points;</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator;</td>
</tr>
<tr>
<td>voxel</td>
<td>Volumetric pixel;</td>
</tr>
<tr>
<td>WYSIWYG</td>
<td>What You See Is What You Get;</td>
</tr>
<tr>
<td>Bio-Formats, 2, 4, 59</td>
<td></td>
</tr>
<tr>
<td>Bio-formats see LOCI, 11, 54</td>
<td></td>
</tr>
<tr>
<td>Bio7, 3</td>
<td>BioImageXD, 4</td>
</tr>
<tr>
<td>Bitwise operations, 117</td>
<td></td>
</tr>
<tr>
<td>Black background, 78, 112, 116</td>
<td></td>
</tr>
<tr>
<td>Blend, 20, 100–102</td>
<td></td>
</tr>
<tr>
<td>Boned, 3</td>
<td>Bounding box, 65</td>
</tr>
<tr>
<td>Brightness/Contrast, 75, 76, 78, 108</td>
<td></td>
</tr>
<tr>
<td>Brush, 37</td>
<td>Brush selection tool, 30</td>
</tr>
<tr>
<td>Bug (reporting) see also Debug, 7</td>
<td></td>
</tr>
<tr>
<td>Calibration, 140</td>
<td></td>
</tr>
<tr>
<td>Optical density, 138, 152</td>
<td></td>
</tr>
<tr>
<td>Spatial, 82, 138, 152</td>
<td></td>
</tr>
<tr>
<td>Calibration bar, 152</td>
<td></td>
</tr>
<tr>
<td>Caliper length see Feret, 136</td>
<td></td>
</tr>
<tr>
<td>Canvas, 81</td>
<td></td>
</tr>
<tr>
<td>Capture image, 159</td>
<td></td>
</tr>
<tr>
<td>CellProfiler see Interoperability, 4</td>
<td></td>
</tr>
<tr>
<td>CellCounter plugin, 33, 34</td>
<td></td>
</tr>
<tr>
<td>CellProfiler see Interoperability, 4</td>
<td></td>
</tr>
<tr>
<td>Center of mass, 136</td>
<td></td>
</tr>
<tr>
<td>Centroid, 136</td>
<td></td>
</tr>
<tr>
<td>Channels, 82, 96</td>
<td></td>
</tr>
<tr>
<td>CIE, 80</td>
<td>CIE Lab, 14, 79</td>
</tr>
<tr>
<td>Circularity, 132, 136</td>
<td></td>
</tr>
<tr>
<td>Clipboard, 48, 60</td>
<td></td>
</tr>
<tr>
<td>Clojure, 26</td>
<td></td>
</tr>
<tr>
<td>CodeBar, 25</td>
<td></td>
</tr>
<tr>
<td>Color</td>
<td></td>
</tr>
<tr>
<td>Background, 36, 84</td>
<td></td>
</tr>
<tr>
<td>Balance, 77</td>
<td></td>
</tr>
<tr>
<td>Blindness, 15</td>
<td></td>
</tr>
<tr>
<td>Composites, 15</td>
<td></td>
</tr>
<tr>
<td>Convolution, 15</td>
<td></td>
</tr>
<tr>
<td>Foreground, 36, 84</td>
<td></td>
</tr>
<tr>
<td>Models, 14</td>
<td></td>
</tr>
<tr>
<td>Quantization, 14, 74</td>
<td></td>
</tr>
<tr>
<td>Quantization see Heckbert quantization, 74</td>
<td></td>
</tr>
<tr>
<td>Separation see Color Deconvolution, 15</td>
<td></td>
</tr>
<tr>
<td>Settings, 69</td>
<td></td>
</tr>
<tr>
<td>Threshold, 79</td>
<td></td>
</tr>
<tr>
<td>Color filter array, 14</td>
<td></td>
</tr>
<tr>
<td>Color palette see LUT, 14</td>
<td></td>
</tr>
<tr>
<td>Color Picker, 36</td>
<td></td>
</tr>
<tr>
<td>Combine, 92</td>
<td></td>
</tr>
<tr>
<td>Command key, 8</td>
<td></td>
</tr>
<tr>
<td>Command launcher, 157</td>
<td></td>
</tr>
<tr>
<td>Compile, 71, 161</td>
<td></td>
</tr>
<tr>
<td>Concatenate, 92</td>
<td></td>
</tr>
<tr>
<td>Contextual Menu, 41</td>
<td></td>
</tr>
<tr>
<td>Control Panel, 157</td>
<td></td>
</tr>
<tr>
<td>Conversions, 70, 126</td>
<td></td>
</tr>
<tr>
<td>Convex hull, 63</td>
<td></td>
</tr>
<tr>
<td>Convolution, 106, 110, 122, 123</td>
<td></td>
</tr>
<tr>
<td>Coordinates, 28, 33, 58, 107, 137, 145</td>
<td></td>
</tr>
<tr>
<td>Coronal see Orthogonal views, 88</td>
<td></td>
</tr>
<tr>
<td>Counting objects, 33, 34, 106</td>
<td></td>
</tr>
<tr>
<td>Crop, 96, 127</td>
<td></td>
</tr>
<tr>
<td>CSV, 42, 53, 58</td>
<td></td>
</tr>
<tr>
<td>CT, 77, 90</td>
<td></td>
</tr>
<tr>
<td>Cursor see Pointer, 45, 72</td>
<td></td>
</tr>
<tr>
<td>Curve fitting, 147</td>
<td></td>
</tr>
<tr>
<td>Built-in functions, 148</td>
<td></td>
</tr>
<tr>
<td>Data mining see TrakEM2, 4</td>
<td></td>
</tr>
<tr>
<td>De-interleave, 93</td>
<td></td>
</tr>
<tr>
<td>Dead pixels, 110</td>
<td></td>
</tr>
<tr>
<td>Debug, 44, 71, 72, 114, 115, 158</td>
<td></td>
</tr>
<tr>
<td>Deconvolution see Convolution, 106</td>
<td></td>
</tr>
<tr>
<td>Deflate see Zip compression, 10</td>
<td></td>
</tr>
<tr>
<td>Delaunay see Voronoi, 115</td>
<td></td>
</tr>
<tr>
<td>Despeckle, 110</td>
<td></td>
</tr>
<tr>
<td>Developer Menu, 37</td>
<td></td>
</tr>
<tr>
<td>Developer resources, 163</td>
<td></td>
</tr>
<tr>
<td>DICOM, 10, 72, 77</td>
<td></td>
</tr>
<tr>
<td>Digitized graphs, 146</td>
<td></td>
</tr>
<tr>
<td>Dilation, 113</td>
<td></td>
</tr>
<tr>
<td>Dirichlet see Voronoi, 115</td>
<td></td>
</tr>
<tr>
<td>Display range, 74</td>
<td></td>
</tr>
<tr>
<td>Distribution, 134</td>
<td></td>
</tr>
<tr>
<td>Divide see Math, 117</td>
<td></td>
</tr>
<tr>
<td>Divide images see Image calculator, 127</td>
<td></td>
</tr>
<tr>
<td>Dot blot analysis, 137</td>
<td></td>
</tr>
<tr>
<td>Downsizing, 80, 97</td>
<td></td>
</tr>
<tr>
<td>DPI, 55, 59</td>
<td></td>
</tr>
<tr>
<td>Drag &amp; Drop, 49</td>
<td></td>
</tr>
<tr>
<td>Duplicale, 93, 96</td>
<td></td>
</tr>
<tr>
<td>Eclipse, 25</td>
<td></td>
</tr>
<tr>
<td>Editor, 43</td>
<td></td>
</tr>
</tbody>
</table>
Your help is needed to improve ImageJ. Even if you are not a programmer, your participation is important:

- Are you a skilled ImageJ user?
  You might want to help out with the documentation effort: Write a FAQ, How-To, Tutorial or Video Tutorial on the ImageJ Documentation Portal; Help us updating the ImageJ User Guide; Share the add-ons you may have created with the community; Subscribe the mailing list and help answering the questions raised by other users.

- Are you know knowledgeable in image processing?
  Please join the mailing list and participate in the discussions. You could also write a Tutorial on the ImageJ Documentation Portal.

- Do you have a strong scientific background?
  Frequently, the most demanding tasks in scientific image processing relate to experimental design. Even if you do not have special expertise in image processing, by participating on the mailing list discussions, your experience will be valuable to others.

- Do you want ImageJ to improve?
  You can report bugs or request new features using the mailing list.

- Do you have experience in graphic/web design?
  If you are able to to improve the look and feel of the guide we welcome your skills.

The ImageJ Icon

The Hartnack microscope (ca. 1870’s) depicted on the front page inspired the ImageJ icon for Mac OS X. It is based on a photograph by Tom Grill at arsmachina.com.

Edmund Hartnack (1826–1891) was a renowned microscope manufacturer that pioneered the use of correction collars in water-immersion lenses and the adoption of the substage condenser.¹ The precision and robustness of Hartnack optics played a pivotal role in the groundbreaking research by the Nobel laureates Robert Koch², Camillo Golgi³ and Santiago Ramón y Cajal⁴.

²Brock, TD. Robert Koch. A life in medicine and bacteriology. *ASM*, 1999
Document Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>September/October 2012</td>
<td>Reviewed by Michael Schmid</td>
</tr>
<tr>
<td>July 2012</td>
<td>Updated for v 1.46r</td>
</tr>
<tr>
<td></td>
<td>Reviewed by Barry DeZonia</td>
</tr>
<tr>
<td>June 2012</td>
<td>Updated for v 1.46p</td>
</tr>
<tr>
<td></td>
<td>Second major revision with new sections on overlays, 3D volumes, sub-pixel selections, curve fitting and interface customizations</td>
</tr>
<tr>
<td></td>
<td>Improvements in layout and browsability</td>
</tr>
<tr>
<td>September 2011</td>
<td>Updated for v 1.45</td>
</tr>
<tr>
<td></td>
<td>Deeply revised edition with several new sections in Parts I–IV</td>
</tr>
<tr>
<td></td>
<td>Available as printable booklets</td>
</tr>
<tr>
<td></td>
<td>Redesigned HTML version</td>
</tr>
<tr>
<td>January 2011</td>
<td>Updated for v 1.44</td>
</tr>
<tr>
<td></td>
<td>New sections on advanced ImageJ usage, Fiji scripting, command line</td>
</tr>
<tr>
<td></td>
<td>usage and interoperability with other software packages</td>
</tr>
<tr>
<td>May 2010</td>
<td>First HTML version</td>
</tr>
<tr>
<td>April 2010</td>
<td>First edition: v 1.43</td>
</tr>
</tbody>
</table>

Acknowledgments

We are extremely grateful to Alex Fernández for his wonderful eLyXer, Alex Gorbatchev for SyntaxHighlighter, Johannes Schindelin for assistance with the Git repository, Norbert Vischer for critical corrections and Michael Schmid and Barry DeZonia for thorough revisions. We are also thankful to all of those who submitted criticisms, suggestions and encouragement to update this edition. But above all, our thanks go to the extraordinary ImageJ community for its dedication to the project.