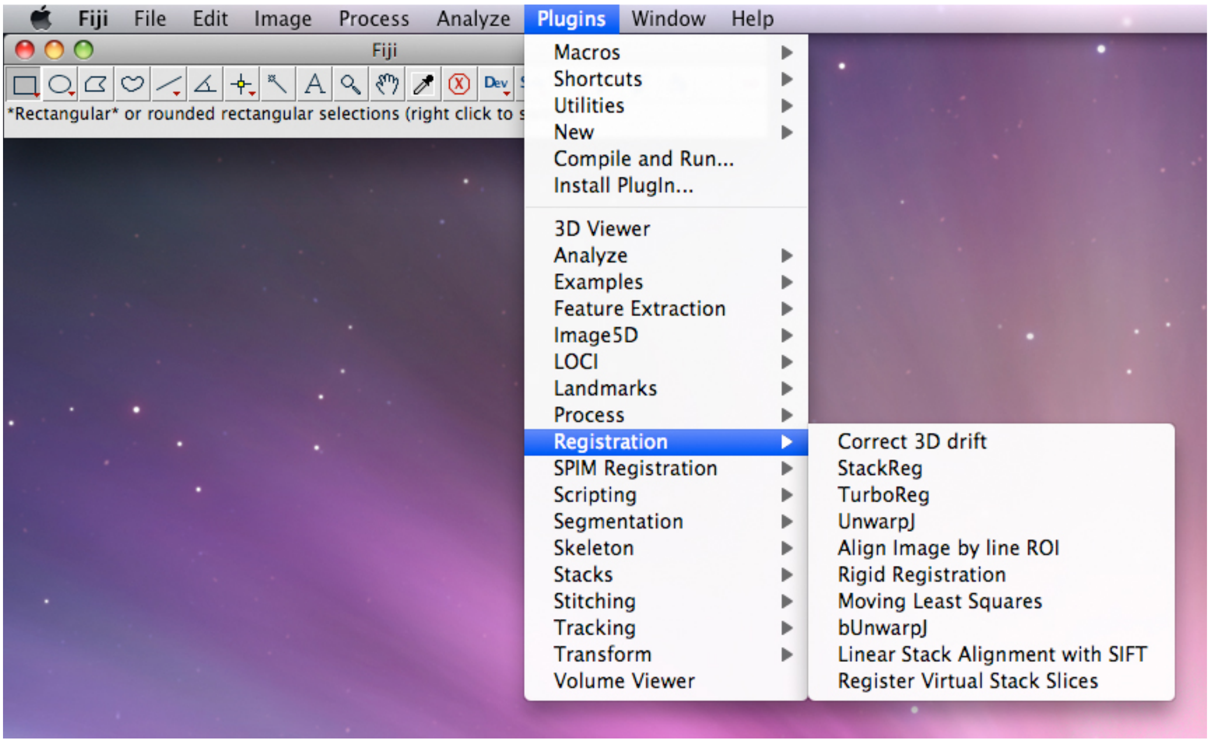
Nature Methods

Fiji: an open-source platform for biological-image analysis

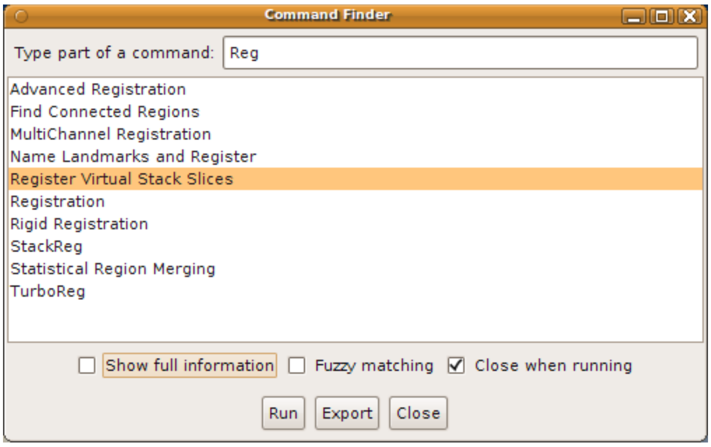
Johannes Schindelin, Ignacio Arganda-Carreras, Erwin Frise, Verena Kaynig, Mark Longair, Tobias Pietzsch, Stephan Preibisch, Curtis Rueden, Stephan Saalfeld, Benjamin Schmid, Jean-Yves Tinevez, Daniel James White, Volker Hartenstein, Kevin Eliceiri, Pavel Tomancak & Albert Cardona

Note: Supplementary Videos 1–4 are available on the Nature Methods website.

Supplementary Figures



Command Launcher

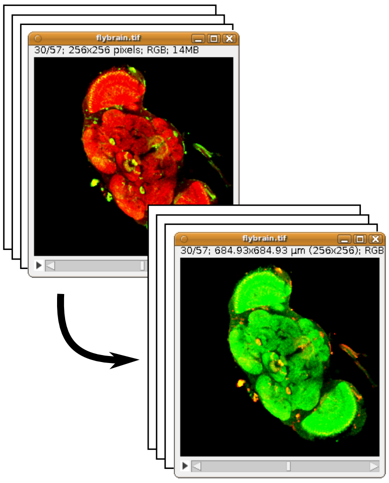


Type part of a command: Reg

□ Show full information □ Fuzzy matching ☑ Close when running

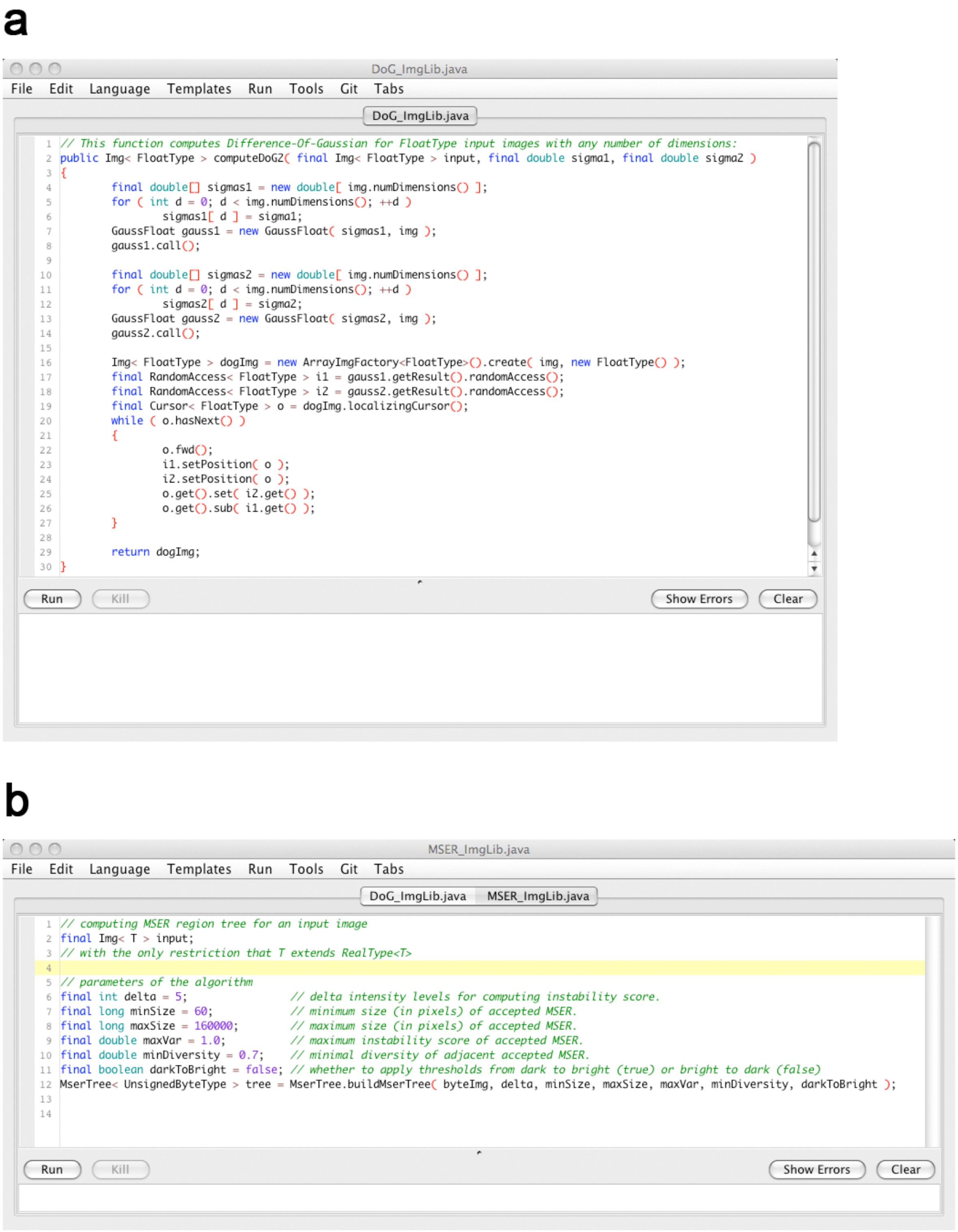
Supplementary Figure 1: Fiji commands The Fiji application packs several hundred commands, organized by category. Each command executes a plugin. *Above*, a snapshot of the *Registration category*. The separation of hundreds of commands into categories relevant to biological researchers greatly aids new users in the discovery of such commands. Most Fiji plugins appear as commands under the *Plugins Menu*; others are inserted into the other menus inherited from ImageJ. For best ease of use, a Command Finder command (*below*), launched via the keybinding *l* (lowercase L), provides the means to find and launch any command with a real-time search that responds to keystrokes. Where a plugin appears in the menu is specified in the *plugins.config* file packed in the plugin's jar file. Any plugins that are not part of Fiji but which adhere to this ImageJ convention will integrate transparently with the Fiji menus.

Batch-processing: swap red and green channels for a set of image stacks


flybrain.tif
30/57; 256x256 pixels; RGB; 14MB
flybrain.tif
30/57; 684.93x684.93 µm (256x256); RGB

```
 1 import os
 2
 3 def getDir (title) :
 4   """ Show a dialog to choose a directory. """
 5   return DirectoryChooser (title) .getDirectory ()
 6
 7 sourceDir = getDir ("Source directory")
 8 targetDir = getDir ("Target directory")
 9
10 if sourceDir == targetDir:
11   IJ.showMessage ("Source and target directories are the same!")
12 else:
13   for filename in os.listdir (sourceDir) :
14     if not filename.endswith (".tif") : continue
15     img = ImgLib.open(sourceDir + filename)
16     swapped = RGBA (Green (img) , Red (img) , Blue (img) , Alpha (img)) .asImage ()
17     swapped.setName (img.getName ())
18     wasSaved = ImgLib.save (swapped, targetDir + filename)
19     print "Saved %s: %s" % (filename, str(wasSaved))
```

Supplementary Figure 2: Scripting in Fiji. An example of a simple Jython script that achieves relatively simple task of swapping the channels (using *ImgLib* lines 15 and 16) but is able to apply it to a directory of images using file manipulation commands inherent to the scripting language.



```java
// This function computes Difference-Of-Gaussian for FloatType input images with any number of dimensions:
public Img< FloatType > computeDoG2( final Img< FloatType > input, final double sigma1, final double sigma2 )
{
        final double[] sigmas1 = new double[ img.numDimensions() ];
        for ( int d = 0; d < img.numDimensions(); ++d )
                sigmas1[ d ] = sigma1;
        GaussFloat gauss1 = new GaussFloat( sigmas1, img );
        gauss1.call();

        final double[] sigmas2 = new double[ img.numDimensions() ];
        for ( int d = 0; d < img.numDimensions(); ++d )
                sigmas2[ d ] = sigma2;
        GaussFloat gauss2 = new GaussFloat( sigmas2, img );
        gauss2.call();

        Img< FloatType > dogImg = new ArrayImgFactory<FloatType>().create( img, new FloatType() );
        final RandomAccess< FloatType > i1 = gauss1.getResult().randomAccess();
        final RandomAccess< FloatType > i2 = gauss2.getResult().randomAccess();
        final Cursor< FloatType > o = dogImg.localizingCursor();
        while ( o.hasNext() )
        {
                o.fwd();
                i1.setPosition( o );
                i2.setPosition( o );
                o.get().set( i2.get() );
                o.get().sub( i1.get() );
        }

        return dogImg;
}
```

```java
// computing MSER region tree for an input image
final Img< T > input;
// with the only restriction that T extends RealType<T>

// parameters of the algorithm
final int delta = 5;                // delta intensity levels for computing instability score.
final long minSize = 60;            // minimum size (in pixels) of accepted MSER.
final long maxSize = 160000;        // maximum size (in pixels) of accepted MSER.
final double maxVar = 1.0;          // maximum instability score of accepted MSER.
final double minDiversity = 0.7;    // minimal diversity of adjacent accepted MSER.
final boolean darkToBright = false; // whether to apply thresholds from dark to bright (true) or bright to dark (false)
MserTree< UnsignedByteType > tree = MserTree.buildMserTree( byteImg, delta, minSize, maxSize, maxVar, minDiversity, darkToBright );
```

Supplementary Figure 3: Examples of ImgLib code. (**a**) Screenshot of the *Script Editor* with ImgLib code that computes Difference of Gaussian (DoG) for float type input images with any number of dimenstions. (**b**) Screenshot of the *Script Editor* with ImgLib code that computes Maximally Stable Extremal Regions (MSER) for float type input images with any number of dimenstions.

Plugins	
Analysis	
3D Objects Counter	Quantification of connected components in 3D
Analyze Skeleton	Calculate statistics on skeletonized objects in binary images
Colocalization	Calculate Pearson and Manders coefficients and make scatter plots for colocalization analysis
Coloc 2	Calculate Pearson, Manders and other colocalization coefficients and make scatter plots for colocalization analysis
Delaunay/Voronoi	Provide a triangulation or Voronoi diagram for a given set of points
Directionality	Quantify direction in images with oriented textures
Exact Euclidean Distance	A linear-time exact algorithm for multi-dimensional Distance Transform
FlowJ	Optical flow analysis
IsoData Classifier	Histogram-based multi-level thresholding
Local Thickness	Quantification of linear structures' thickness in two or three dimensions
PIV Analyser	Particle Image Velocimetry, an optical flow algorithm
Shape Index Map	Local surface shape classification
Shape-based Averaging	Make averages of multiple images preserving shapes
Skeletonize 3D	Skeletonize 3D objects in binary image stacks
Surfaceness	Study the morphology of volumetric data containing surface-like structures
Temporal Color Coder	Color-coded maximum intensity projection of time series
TopoJ	Analysis of surface topology
Tubeness	Enhance tube-like structures in volumetric data
Image transformation	
3D Morphology	Morphological operations on 3D binary images
Anisotropic Diffusion	Edge-preserving blurring
Bilateral Filter	An edge-preserving Gaussian blur
Colour Deconvolution	Identification of fluorophores by color
Differentials	Differential operators
Dynamic Reslice	Make a kymograph along a line of interest, updated interactively
Extended Depth Field	Stack focusing, i.e. make in-focus images from stacks where each (x,y) coordinate is in focus in only one or two z slices

Gray Morphology	Morphological operations on grayscale images
Radial Reslice	Transform to polar coordinates
TransformJ	Arbitrary affine transformations
Image Arithmetic	
Calculator Plus	Advanced arithmetic operations on two images
Image Expression Parser	Application of free-form terms (e.g. $A + 2 * B - gauss(A, 0.5)$, where A denotes an image).
Image enhancing	
CLAHE	Contrast Limited Adaptive Histogram Equalization method for enhancing the local contrast of an image.
Frangi Vesselness	Frangi's filter to enhance linear structures in noisy images
Kalman Stack Filter	Kalman's temporal denoising method
Kuwahara Filter	Kuwahara's denoising filter
Retinex	Perception-based contrast and color enhancing
ROF Denoiser	Image denoising based on the paper by Rudin, Osher and Fatemi
Tracking	
Manual Tracking	Manual tracking of multiple objects in time series
MTrack2	Tracking of multiple objects in time series
ToAST	Automated tracking of sporozoids in time series
Segmentation	
Auto Threshold	Reusable histogram-based global and local threshold calculation
Colour Threshold	Threshold segmentation based on colors, e.g. selecting ranges of hues and ranges of saturation
FeatureJ	Classical differential image features
GraphCut	Graph-based segmentation
Lasso and Blow Tool	interactive tools adapting the selection to the pixel intensities
Level Sets	Segmentation based on Level Sets (Active Contours)
Linear Kuwahara	Enhancement of linear structures based on the Kuwahara filter
Robust Adaptive Threshold Selection	Local gradient-based threshold selection
SIOX Segmentation	Color segmentation based on example segmentations

Simple Neurite Tracer	Semi-interactive segmentation of fiber-like structures, as well as analysis of the resulting data.
Snakuscule	Circular active contours
Statistical Region Merging	Automatic Region Merging based on a statistical test of the intensity differences
Trainable Segmentation/Weka Segmentation	Advanced machine learning technique to extract a segmentation model from example segmentations
Registration	
bUnwarpJ	Automatic bijctive elastic registration of 2D images
Align images by a common line	Rigid registration according to line selections in two images marking the same structure
Moving Least Squares	Landmark-based registration, interpolating the transformation as rigidly as possible
Register Virtual Stack Slices	Register unaligned slices of one stack, with minimal memory requirements
Rigid Registration	Automatic and manual rigid registration of two 3D images
SPIM Registration	Register multiple 3D images of the same sample, recorded from different angles
StackReg	Automatic 3D alignment of two image stacks
TurboReg	Automatic 2D alignment of two images
VIB Protocol	The Virtual Insect Protocol aims to facilitate anatomical studies from multiple samples
Stitching	
MosaicJ	Semi-automatic stitching of 2D image mosaics
Stitching 2D/3D	Automatic stitching of 3D image mosaics
Annotation	
Arrow Tool	Draw and adjust arrows
Series Labeler	Versatile time stamper
Stack Manipulation	Reorder, recombine, interleave stacks and other stack-specific operations
Sync Win	Track current mouse coordinates in other images
Visualization	
3D/4D Viewer	Using Java3D for hardware-accelerated 3D/4D display of images
Dichromcay	Simulating color blindness
Interactive 3D Surface Plot	Interactively display a 2D image as a height field
Multi Kymograph	Kymographs (intensity over time plots) from stacks and line selections
Panorama equirectangular view	Visualize panorama images in a pannable view

Record Screen	Make a movie from a window, e.g. for screencasts
Replace Red by Magenta	A tool to fix color blind-unfriendly red/green images
Video Editing	Basic video editing tools for editing screencasts
View 5D	Display up to 5D volumetric datasets, offering convenient data inspection, basic particle tracking, and more
Volume Viewer	Volume rendering without requiring hardware acceleration
Input/output	
Amira Reader/Writer	Load and save 3D images in the Amira file format
Analyze Reader/Writer	Load and save files in the Analyze file format
Animated GIF Reader/Writer	Load and save image stacks as animated GIF files
Bio-Formats	Read/write support for 100 file formats commonly used by microscope vendors' software
Biorad Reader/Writer	Load and save images in the format written by Biorad's software
DF3 Reader/Writer	Load and save files in the Povray texture format
DM3 Reader	Import files produced by the Gatan Digital Micrograph
IPLab Reader	Import files in the IPLab format
Export EPS	Write images as .eps files for publication
LSM Reader/Toolbox	Special-purpose I/O for the file format used by Zeiss
Nrrd Reader	Reader for the Nearly Raw Raster Data format
PDF Reader	Render a PDF document into an image stack
PDF Writer	Make a .pdf file from one or more images
Reconstruct Reader	Import Reconstruct projects as TrakEM2 projects
SVG Reader	Render SVG into a raster image
Scripting	
Beanshell, Clojure, Javascript, JRuby, Jython	Support for scripting and interactive evaluation of commands
Scripting Examples	Fiji comes with a large number of examples that can be adjusted to one's own needs
Java as a "script" language	For rapid prototyping, Fiji can treat simple Java classes as if they were scripts
Script Editor	A versatile editor helping to write scripts
Working with Fiji	

Automatic class version translator	Make it possible to run plugins compiled for Java 1.6 on older Java versions (e.g. on MacOSX versions prior to 10.6)
Bug Submitter	Provide an easy and convenient way to submit a bug report
Context help	Open the Fiji Wiki page corresponding to a given menu item
Fiji Updater	Effortless and user-friendly way to stay up-to-date
IJ Robot	Automate mouse clicks and keyboard presses
Multiple Image Processor	Batch process images in a directory output to another directory
Upload Sample Image	Allow users to upload large sample images for use by the Fiji developers
Interoperability	
Miji	Provide a user-friendly way to use Fiji from within Matlab
Miscellaneous	
ImageJ/ImageJA	A custom ImageJ version that frequently contains cutting edge features which are then integrated into ImageJ
QuickPALM	Software support for Photo-Activated Light Microscopy (superresolution microscopy)
RandomJ	Generate artificial noise adhering to given parameters
Thread killer	Stop runaway processes
TrakEM2	Work with large mosaics, e.g. obtained via EM (segmentation, stitching, registration and much more)
Developing Fiji	
Eclipse Netbeans IntelliJ support	Fiji has support for popular development environments
Tutorial Maker	A very fast way to make tutorials in Fiji and upload them to the Fiji Wiki

Table 1: Overview of Fiji plugins The table list all plugins currently provided by the Fiji Updater system together with a short description. Plugins developed specifically for Fiji are highlighted in blue (77 of 111 total). The remaining plugins are inherited or selected from ImageJ.