Overview

Quantitating the cell: turning images into numbers with ImageJ

Ellen T Arena,[1,2] Curtis T Rueden,[2] Mark C Hiner,[2] Shulei Wang,[3] Ming Yuan[1,3] and Kevin W Eliceiri[1,2]*

Modern biological research particularly in the fields of developmental and cell biology has been transformed by the rapid evolution of the light microscope. The light microscope, long a mainstay of the experimental biologist, is now used for a wide array of biological experimental scenarios and sample types. Much of the great developments in advanced biological imaging have been driven by the digital imaging revolution with powerful processors and algorithms. In particular, this combination of advanced imaging and computational analysis has resulted in the drive of the modern biologist to not only visually inspect dynamic phenomena, but to quantify the involved processes. This need to quantitate images has become a major thrust within the bioimaging community and requires extensible and accessible image processing routines with corresponding intuitive software packages. Novel algorithms both made specifically for light microscopy or adapted from other fields, such as astronomy, are available to biologists, but often in a form that is inaccessible for a number of reasons ranging from data input issues, usability and training concerns, and accessibility and output limitations. The biological community has responded to this need by developing open source software packages that are freely available and provide access to image processing routines. One of the most prominent is the open-source image package ImageJ. In this review, we give an overview of prominent imaging processing approaches in ImageJ that we think are of particular interest for biological imaging and that illustrate the functionality of ImageJ and other open source image analysis software.

How to cite this article:
WIREs Dev Biol 2017, 6:e260. doi: 10.1002/wdev.260

INTRODUCTION

Technical breakthroughs are transforming the world of scientific imaging. This is most evident in optical microscopy, where advanced, multidimensional imaging modalities are providing higher temporal and spatial resolutions than ever before from an array of sophisticated bio-inspired *in vitro* systems[41,44] to tracking active cells in live animals *in vivo*.[21,28,62] This has provided researchers a real-time view of complex biological processes, including oncogenic signaling,[17] tumor metabolism,[61] and vesicular trafficking,[12] as well as the complex architecture of a developing organism.[14,24] Imaging approaches provide the means to visualize such events both spatially and temporally, but there also remains a great need to quantitatively measure these events. Computational techniques to do this now constitute an emerging field, bioimage informatics,[45] with a great number of approaches that allow scientists to extract quantitative data from their images.[20] However, with such a diverse selection of techniques, procedures, and tools, it can be difficult for biologists to determine where to best begin their analysis. When examining complex biological processes,

*Correspondence to: eliceiri@wisc.edu

[1]Morgridge Institute for Research, Madison, WI, USA

[2]Laboratory for Optical and Computational Instrumentation, University of Wisconsin at Madison, Madison, WI, USA

[3]Department of Statistics, University of Wisconsin at Madison, Madison, WI, USA

Conflict of interest: The authors have declared no conflicts of interest for this article.

careful considerations need to be made not only for image acquisition systems, but also for postprocessing and bioimage analysis protocols. There stands before the user a vast collection of computational choices, not only in the algorithms themselves, but also in the various software tools that provide these algorithms to the greater scientific community. Characteristics of an ideal software platform[10,11] will be one designed for science, allowing open inspection and verification, while maintaining the necessary flexibility to support the imaging field's ever-expanding repertoire of techniques and modalities. One such open-source software toolkit is ImageJ.[60] In this article, we discuss a select subset of image analysis techniques available through ImageJ to specifically highlight its utility in scientific image analysis. We hope this presentation of commonly applied techniques will also encourage exploration of other powerful tools available within ImageJ and related open-source packages. We focus on ImageJ applications and use cases that are likely of most interest to the developmental and cell biology fields, but many of the tools and information presented are applicable to other biological domains.

Open-Source Image Processing

While there is merit to commercial bioimage analysis software, this review focuses on open source image processing and one platform in particular: ImageJ. ImageJ is arguably the most widely utilized open source bioimage analysis tool and serves as a good example of the benefits of open source. Open-source equals transparency, providing the ability to inspect, reproduce, and verify, which is absolutely necessary for the scientific process.[10] Computational tools are playing an increasingly significant role in science, as a result, software needs to be transparent so scientists can fully understand the computational methods being applied to their particular biological questions, and so these methods can be expanded and improved.[11] Simply stated, scientific data and methods must be shared to have value, and proprietary tools create implicit barriers to this process.

As was thoroughly discussed by Cardona and Tomancak,[10] the biological research community must engage in collaborations with computer scientists in the area of bioimage analysis to continue the expansion of scientific research. ImageJ epitomizes this union as an open-source tool driven by contributions from collaborating scientists and developers. As demonstrated by the very active ImageJ Forum (http://forum.imagej.net/) to the growing list of update sites (http://imagej.net/Update_Sites), building ImageJ is a true community effort. Therefore, we wish to emphasize not only the role of ImageJ as an image analysis platform and the tools and techniques it makes available to users, but also the role of the community in shaping future directions of open-source tools for scientific bioimaging.

ImageJ and Fiji

From a user standpoint, ImageJ is an open-source image processing platform for multidimensional image data, built for the specific needs of scientific imaging[57] (http://imagej.net/). It is an application for all aspects of image analysis postacquisition, providing functions to load, display, and save images, coupled with a robust repertoire of image processing techniques with dedicated tools for segmentation, data visualization, tracking, lifetime analysis, and colocalization to name only a few biological applications. In general, a key entry point for new users to ImageJ is via the ImageJ website (http://imagej.net). The Introduction section of the ImageJ wiki (http://imagej.net/Introduction) is a perfect resource for beginners to ImageJ and image analysis in general (Table 1). There are also helpful tutorials (http://imagej.net/Category:Tutorials), as well as detailed user guides (http://imagej.net/User_Guides); as well, there is the all important ImageJ Forum (http://forum.imagej.net/) where new users can post questions to the community for assistance in their general or specific image analysis needs.

While ImageJ began as a single, standalone application, it has grown to encompass a broad collection of related software libraries and applications.[57] The original application is known as ImageJ 1.x within the community, whereas the updated, full suite of components now reengineered for *N*-dimensional, scalable analyses is referred to as ImageJ2: a collection of reusable software libraries, extensible plugins/services, and reusable image processing operations. This ImageJ family is a multifaceted project built upon the SciJava framework, a foundational software layer for scientific computing—including image processing, visualization, annotation, workflow execution, and machine learning—which strives to consolidate and reuse code whenever possible.

We will simply use the phrase 'ImageJ' for consistency throughout this review; however, readers should be aware that because of the extensible nature of ImageJ, its specific capabilities can vary based on what is installed in a particular instance of the application. The most reliable mechanism for extension is via update sites (http://imagej.net/Update_Sites),[56] which typically include institution or task-focused

TABLE 1 | Key ImageJ resources

ImageJ Resources	Description	Links
ImageJ wiki	Everything you need to know (and more) about ImageJ	http://imagej.net/Welcome
ImageJ introduction	The home base for basic users on the ImageJ wiki	http://imagej.net/Introduction
Getting started with ImageJ	An introduction to the ImageJ application	http://imagej.net/Getting_Started
ImageJ user guide	Provides a thorough description of ImageJ's basic, built-in functions	http://imagej.net/docs/guide/
Principles of image analysis	Must-read guidelines for effective acquisition and analysis of images	http://imagej.net/Principles
The Fiji cookbook	'Recipes' and techniques for image processing	http://imagej.net/Cookbook
Scripting in ImageJ	Become a power user by writing scripts	http://imagej.net/Scripting
ImageJ workshops	Some key workshops include an Introduction to Fiji, Segmentation in Fiji, and Scripting with Fiji	http://imagej.net/Workshops
The ImageJ forum	The recommended way to get help. A very active community; a rich, modern interface; public, archived discussions	http://forum.imagej.net/

collections of plugins. Fiji (Fiji is Just ImageJ) is the largest and best-known distribution of ImageJ,[56] providing a curated collection of preinstalled plugins. Many of the techniques we will highlight here have dedicated plugins in Fiji; thus, we recommend new users start with the Fiji update site enabled, as it is in the default Fiji download. By presenting tools and techniques freely available in this open-source package, we hope readers will feel empowered to immediately apply these ImageJ-based techniques to their own biological image datasets.

COMMON IMAGE ANALYSIS TECHNIQUES

There are hundreds of image analysis routines one can apply within any major analysis toolkit, particularly within such an extensible program as ImageJ. Here, we have chosen a few representative analysis techniques and their related ImageJ-based tools that are commonly employed in cellular and developmental biology. Each tool has robust, active development in ImageJ, and together they represent a subset of the power of the unified ImageJ platform. The Techniques page (http://imagej.net/Techniques) on the ImageJ wiki (http://imagej.net) documents the technical aspects of each analysis approach. Our aim here is to introduce users to these image analysis techniques, revealing their biological applicability and providing a clear path to their use within ImageJ.

Segmentation

Understanding cellular morphological features and subcellular structures is key to many biological studies. The size and shape of individual cells and subcellular features can be indicative of physiological state; for example, ImageJ has been used to measure changes in large-scale cell shape, revealing condensation of chromatin and a dramatic effect on cell proliferation.[67] Proper delineation of cells and/or subcellular features—i.e., image segmentation, the process of dividing regions of a digital image to delineate objects or boundaries within—is an incredibly informative and powerful image analysis technique. It is also often the foundation of many subsequent analyses, including cell tracking, lifetime and colocalization measurements, and so on. Essentially, individual pixels are grouped to ensure that pixels with analogous characteristics are similarly labeled.

There exist numerous methods to employ when segmenting images. The selection of an appropriate method depends on the nature of the acquired image(s), as there exists extensive variability in not only the biological samples themselves, but also in the microscopic methods utilized for image acquisitions leading to variations in data complexity. Segmentation techniques are either noncontextual or contextual, the latter of which take into account those neighboring pixels sharing similar gray levels at close spatial locations.

Flexible Segmentation Workflows

Flexible segmentation workflows are user-defined. They can vary depending on the specific datasets to be processed. Steps for such segmentation include (1) preprocessing of images via selection of filters to best facilitate subsequent thresholding. Using ImageJ, such filters can include Background Subtraction, which uses a rolling ball algorithm to correct uneven

background,[64] Gaussian Blur, and Find Edges, which uses a Sobel edge detector to detect sudden changes in intensity levels across an image. Once preprocessing steps have been carried out, (2) thresholds can then be applied. Ideally, the Auto Threshold plugin is used to ensure reproducibility and the removal of user inconsistencies via manual manipulation of thresholds. This particular plugin binarizes 8- and 16-bit images via global thresholding methods, selected from Huang, Intermodes, Li, Mean, Otsu, Yen, and more. Local threshold methods can also be used via the Auto Local Threshold plugin. Once a threshold has been set within an image, it can be used to (3) create a binary mask. Based on the threshold set and the image itself, some areas of the image may be over- or under-saturated. In these situations, the Dilate or Erode operations can be used to either grow or remove pixels from saturation, respectively. The mask can then be used to (4) create and transfer a selection from the mask to the original image, and then finally, (5) resulting datasets can be analyzed and measurements taken—e.g., via the Analyze Particles command of ImageJ. All of these processes can be assembled readily into scripts to allow for the creation of an automatic analysis workflow and batch processing. There are several extensive analysis toolkits available to users through ImageJ that contain various segmentation and other analysis workflows, including BAR (http://imagej.net/BAR) and BioVoxxel (http://www.biovoxxel.de/). These toolkits substantially extend ImageJ's own toolbox, providing users powerful analysis tools within well-documented and -maintained packages.

Trainable Weka Segmentation

The Trainable Weka Segmentation (TWS) plugin (http://imagej.net/Trainable_Weka_Segmentation) is a machine-learning tool that leverages a limited number of user-guided, manual annotations in order to train a classifier and segment the remaining data automatically[2] (Figure 1). The TWS plugin has been used a great deal in automatic tissue segmentation; for example, it was used to develop a fully automated tissue segmentation of neck–chest–abdomen–pelvis computed tomography (CT) examinations for pediatric and adult CT protocols.[48] This plugin is *trainable* in that it can learn from user input and apply similar tasks on unknown datasets. It leverages the *Weka* library[27] (http://www.cs.waikato.ac.nz/ml/weka/), an extensive collection of machine learning algorithms, tools, and classifiers for data mining tasks. TWS essentially functions as a bridge between the fields of machine learning and image processing by providing a user-friendly tool to apply and compare pixel-level classifiers. When classifiers are applied to a complete image, every pixel is assigned a class, and the resulting groupings of these classes create a naturally emergent, labeled segmentation. It is particularly powerful in cases where 'classical' segmentation methods are not robust enough for reliable, automated segmentation. Some examples include: the joint segmentation of *Escherichia coli* in brightfield images[32] and the automated segmentation of epithelial stromal boundaries for collagen fiber alignment measurements in H&E tissue samples of human breast carcinoma.[68] TWS classification can also be integrated into larger, flexible segmentation workflows as discussed above, replacing the autothreshold step with machine learning, which ultimately produces the binary mask used in subsequent steps.

Registration

The acquisition of large-scale volumetric image data using newer imaging modalities, including light sheet fluorescence microscopy (LSFM) methods, leads to the generation of multiple views of samples that are collected by either interchanging the roles of the objectives or by rotating the sample. Combining the data from these many views leads to improved 3D image resolution by overcoming poor axial resolution, etc. For example, selective plane illumination microscopy (SPIM)[29] has been used for imaging whole developing organisms, including the teleost fish Medaka (*Oryzias latipes*), zebrafish (*Danio rerio*), and the fruit fly (*Drosophila melanogaster*), with single cell resolution at groundbreaking temporal resolution.[34] However, in order to maximize the full potential of these acquisitions, these multiview datasets needs to be reconstructed via the process of registration. Registration involves the spatial unification of a collection of image data into a shared coordinate system.

Image registration is achieved by use of algorithms to determine image alignment, which can be categorized as either intensity-based or feature-based algorithms. Intensity-based methods examine intensity patterns within images via correlation metrics, whereas feature-based methods determine and compare the positioning of distinct points, lines, or contours for proper alignment. For both, the goal is to spatially transform a target image onto a known reference image. Ultimately, this can involve the application of linear transformations, such as rotation, translation, and affine transforms, as well as

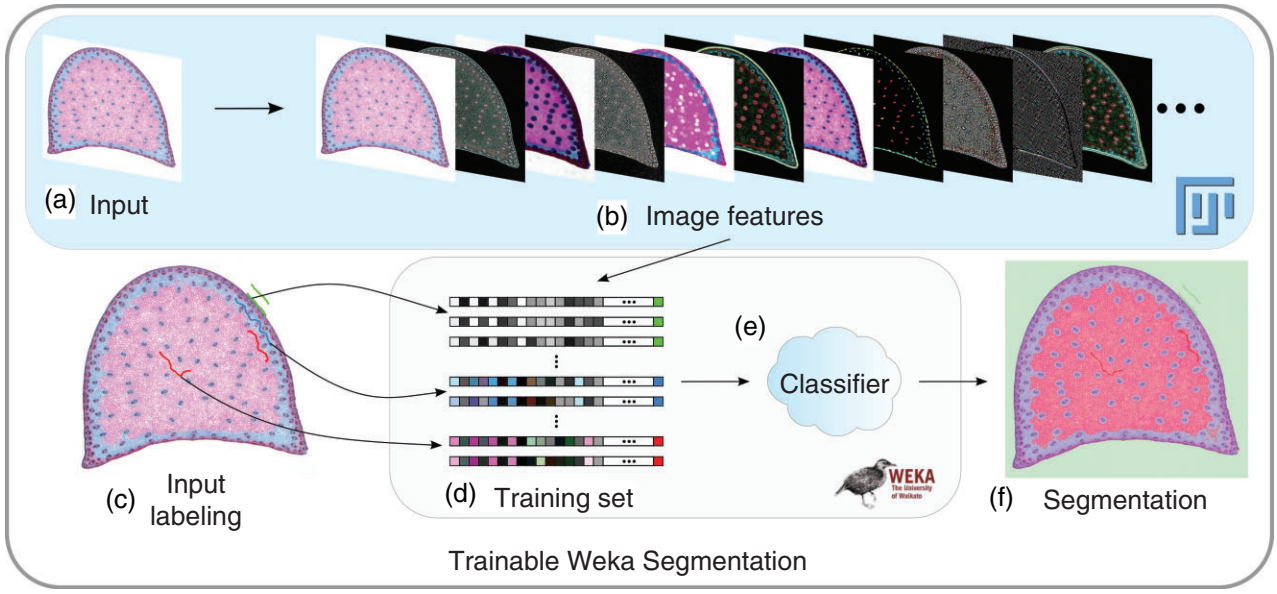

FIGURE 1 | The Trainable Weka Segmentation (TWS) pipeline for pixel classification. Given a sample input image, in this example a maize stem slab acquired using a flat scanner with a resolution of 720 DPI, corresponding to 35.3 μm/pixel (courtesy of David Legland)[69] (a), a user is dependent on the image alone in order to extract features and to properly segment those features; this process can vary greatly depending on the input image (b). Using the power of machine learning, the TWS plugin takes an input image and a set of labels defined by the user to represent feature vectors (c); a WEKA learning scheme is trained on those labels (d) to define and apply a classifier (e) to the remaining image data to properly and automatically segment (f) the image.

'nonrigid' transformations that allow subregion warping of the target image in order to align to the reference.

TrakEM2

TrakEM2 (http://imagej.net/TrakEM2) is an ImageJ plugin for morphological data mining, three-dimensional modeling and image stitching, registration, editing, and annotation[9] (Figure 2). This particular tool features segmentation implementations, including semantic segmentation, volumetric and surface measures, 3D visualization, and image annotation. However, in this particular review, we will focus on its registration capabilities. TrakEM2 has been used in the expeditious reconstruction of neuronal circuits for model systems of both *Drosophila* and *C. elegans*, addressing their systematic reconstruction from both large electron microscopic and optical image volumes.[9] This tool registers floating image tiles to each other using scale invariant feature transform (SIFT) and global optimization algorithms. SIFT uses local features as points of interest to extract those corresponding landmarks so transformations can be calculated and applied within the plugin. TrakEM2 has a very effective, semi-automatic snapping protocol for aligning images. An image can be manually dragged onto another and 'snapped' to it, and after, a subset of pixels is analyzed in order to calculate similarities for best matching the images. TrakEM2 also uses landmarks to align images, where users are able to manually designate reference regions, which the software can use to calculate a corresponding transform. The flexibility and performance of this particular tool provides users with various, powerful registration techniques via an easy-to-use interface.

Multiview Reconstruction

As stated above, LSFM methodologies are transforming the way we image developing organisms. Today, the entire process of *Drosophila* embryogenesis can be imaged without any harm unto the living, developing specimen.[29] The Multiview Reconstruction plugin of ImageJ was developed to specifically handle these types of data, in particular, to register multiview image datasets (http://imagej.net/Multiview-Reconstruction). It is a newer plugin, developed to replace the previous SPIM Registration plugin.[49,50] Multiview Reconstruction allows users to register, fuse, drift-correct, deconvolve, and view multiview image datasets, which can be run as automated workflows (http://imagej.net/Automated_workflow_for_parallel_Multiview_Reconstruction). Although it was specifically designed to deal with LSFM datasets, Multiview Registration can be used for viewing any dataset with 3 or more dimensions, from confocal

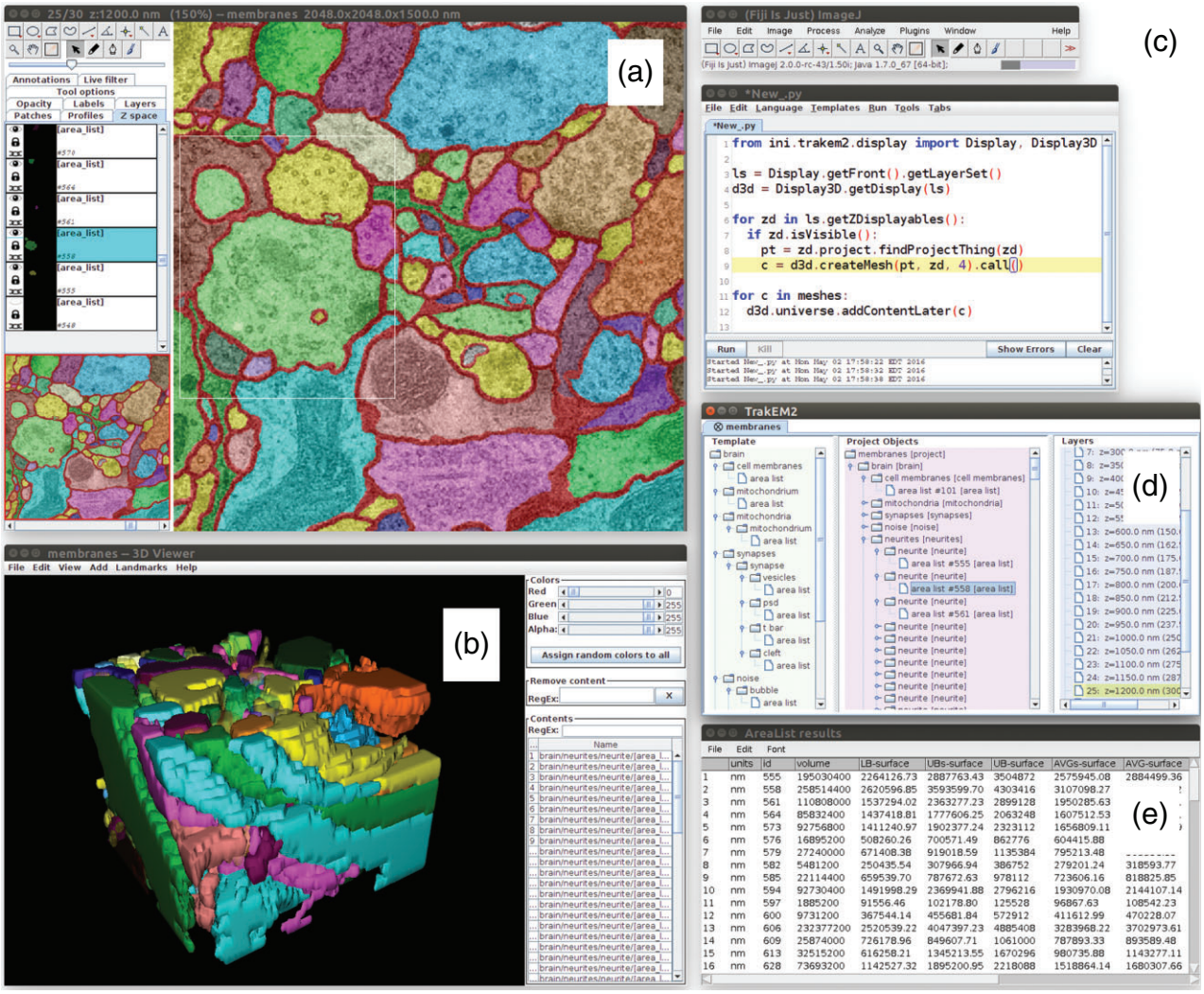

FIGURE 2 | The TrakEM2 plugin assembles 3D volumes and reconstructs, measures, and analyzes structures contained within. In this example, we show a 512 × 512 × 30 px volume at 4 × 4 × 50 nm resolution of a *Drosophila* larval central nervous system,[8] which was registered using automated methods.[54] All cytoplasmic membranes, synapses and mitochondria are segmented (a) using manual (custom brush tools) and semiautomatic methods (Level Sets ImageJ plugin by Erwin Frise). Each element is editable from the UI (a). The reconstructions are hierarchically organized and can be manipulated as groups (d). The volumes can be rendered (b) and measured (e) using ImageJ's 3D Viewer plugin[59] and results table, which provide further means for exporting the data for further processing elsewhere. In addition to the TrakEM2 interface, individual methods can be combined with other techniques from within the script editor (c).

time-series to multichannel 3D stacks. Multiview Reconstruction is fully integrated with the visualization plugin, BigDataViewer, which is to be discussed in more detail in the upcoming section; both tools have been used in research into the early development of the polyclad flatworm *Maritigrella crozieri*, a nonmodel animal.[25]

Visualization

Advanced 3D imaging technologies, such as LSFM methods, including selective/single plane illumination microscopy (SPIM), are changing the way scientists are acquiring live samples, especially in the realm of developmental biology.[30,58] However, with added acquisition dimensionality with space and time and beyond now possible, effective computational tools are needed to handle such large-scale 3D or higher dimension images, their registration and rendering. Such tools are more important than ever before as these datasets are truly in the realm of 'big data' and need to be processed with care to maximize information extraction in a timely manner.

Typically, microscopes acquire images before processing and visualization occurs, which is especially the case for large-scale data acquisitions on the

order of multigigabytes per second acquired over hours and even days. Although only a small subset of this data may be of interest, the complete dataset must be retained until it can be analyzed. This delay between acquisition and data processing poses a detriment to effective use of time and storage, not to mention the image specimens themselves. Steps are currently being taken to bridge this gap by using novel computational tools to quickly provide robust, visual feedback, as most advanced imaging systems only allow for single image planes to be visualized in real time. The sooner *N*-dimensional datasets can be rendered, visualized, and assessed relative to the acquisition process, the more efficiently and effectively science can progress.

BigDataViewer

BigDataViewer is an ImageJ plugin that allows users to navigate and visualize large image sequences from both local and remote data sources[47] (http://imagej.net/BigDataViewer). It was specifically designed for terabyte-sized multiview light-sheet microscopy data, integrated with the SPIM image processing pipeline of ImageJ. The data is handled as a collection of individual sources; for example, in a multiangle, multichannel SPIM dataset, each channel of each angle is considered a source, and within this tool, each source can be displayed and manipulated independently. BigDataViewer has a custom data format, optimized for rapid browsing of image datasets too large to fit entirely in computer memory, maintaining spatial metadata to register sources to the global coordinate system. The modular design of BigDataViewer separates data access, caching, and visualization into separate functions, making it easier to reuse and build upon from other plugins. Its data structures are built on the generic image processing library, ImgLib2, an open-source Java library designed for *N*-dimensional data representation and manipulation for image processing.[46] In the ImageJ software ecosystem, ImgLib2 is the foundation of next-generation image processing operations, allowing natural compatibility with the BigDataViewer.

ClearVolume

ClearVolume is a newer development that is poised to become the flagship 3D volume-rendering tool for ImageJ (http://imagej.net/ClearVolume). For example, BigDataViewer is currently being extended to support volume visualization of massive datasets using ClearVolume for 3D rendering. ClearVolume is an open-source package designed for real-time, GPU-accelerated, 3D + time multichannel visualization and processing[52] (Figure 3). It is a library designed specifically for advanced 3D volume microscopy acquisition methods, including SPIM and DLSM. 3D volumetric stacks can be visualized in real time, as it iteratively estimates data to progressively show more accurate views in order to handle large-scale image volumes. As opposed to waiting for offline postprocessing steps, scientists have immediate views of their data as they acquire it to determine sample quality and image acquisition parameters, and so on. Also, the point spread function (PSF) can be visualized on the acquisition system in 3D to compute image quality in real time during system calibration. ClearVolume was used for all the 3D rendering in a recent study that developed a novel workflow for 3D correlative light and electron microscopy (CLEM) of cell monolayers, specifically to examine the composition of entotic vacuoles.[53]

Tracking

Cells are not immobile entities; biological systems are made up of dynamic processes, including cell membrane dynamics, vesicular trafficking, cytoskeletal rearrangements, focal adhesions, viral and bacterial infections, intracellular transport, and gene transcription and maintenance. Digitally capturing this dimensionality provides a full view of life's processes. As in the case of segmentation studies, where an understanding of cellular morphological features and/or subcellular structures provides greater insight into the physiological state of a cell, tracking takes this one step further by quantifying the dynamic nature of cellular and/or particle movements. For an even more advanced discussion on tracking, in particular within the realm of *Drosophila* developmental biology, please see the extensive review by Jug *et al*.[33]

Tracking of whole cells or subcellular structures, such as organelles, macromolecular complexes, or even single molecules, involves identifying and following these structures over time. Manual tracking is error-prone and impractical, especially when dealing with hundreds or thousands of targets; therefore, computational algorithms need to be employed to effectively and efficiently carry out such tasks. Dozens of software tools are available for tracking, all of which share methods based on the two key components of tracking: spatial and temporal features.[13,40] The spatial component, determined in the *segmentation step*, is the identification and separation of relevant objects from background signal in each frame, as described previously. The temporal component is the association of segmented objects from frame-to-frame, building connections over time; this is the *linking step*. There are a variety of methods that can be applied in this latter step. The nearest-neighbor solution, a local-linking method, links objects from frame-

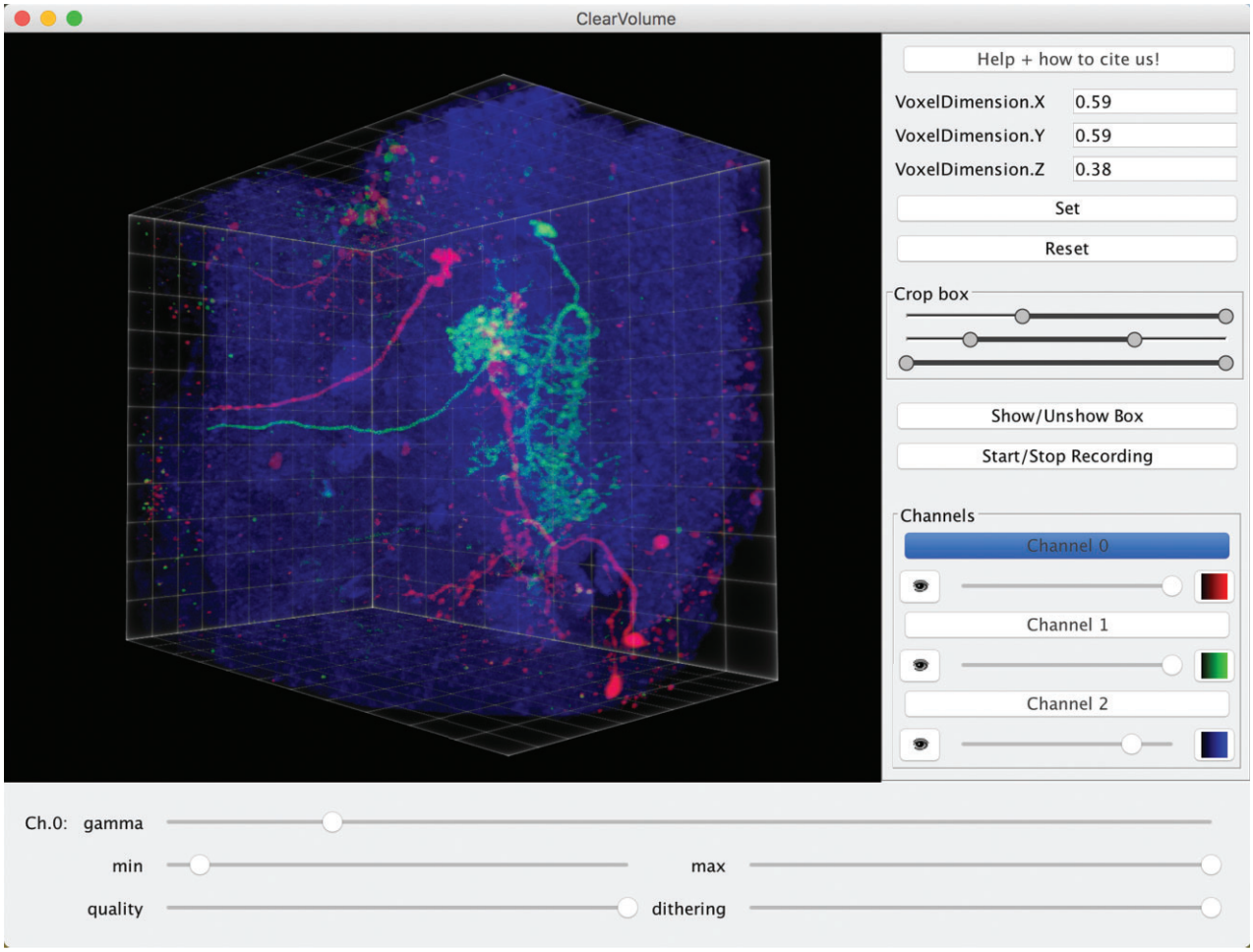

FIGURE 3 | ClearVolume is an open-source multichannel volume renderer. The plugin offers an intuitive user interface with a number of configuration options, including voxel size and axes parallel cropping, as well as setting lookup tables, brightness, contrast, transparency, and render quality for each individual channel. The visualized dataset in this figure shows two *Drosophila* neurons that were labeled with twin-spot mosaic analysis with a repressible cell marker (MARCM) and imaged in the lab of Tzumin Lee at HHMI Janelia Research Campus.[68]

to-frame based on spatial distances, intensity, volume, orientation, or other key features. Multiple global-linking methods exist, including spatiotemporal tracing,[7] graph-based optimization,[31,55] and probabilistic tracking algorithms using Bayesian estimations, which include interacting multiple model (IMM) algorithms based on various models of different biological movement types,[22] approaches based on independent particle filters,[26] and Rao-Blackwellized marginal particle filtering.[63]

Finally, once the particles of interest are successfully segmented and linked over time, a multitude of measurements can be made from the resulting tracks (as reviewed in Ref 40), including total trajectory length, net distance, confinement ratio, mean-squared displacement (MSD), rate of displacement, instantaneous velocity, arrest coefficient, mean curvilinear speed, and the mean straight-line speed. Morphology measurements also consider the object's shape at each time point, including the perimeter, area, circularity, ellipticity, concavity, convexity, and more.

TrackMate

TrackMate is an ImageJ plugin for the automated, semi-automated, and manual tracking of single particles (http://imagej.net/TrackMate). It aims to offer a general solution that works out of the box for end users, through a simple and sensible user interface. TrackMate has been used to carry out lineage analysis on *C. elegans*, examining the effects of light-induced damage during imaging and its potential impact on early development (Figure 4). It operates on time series of 2D or 3D multichannel images and provides several visualization and analysis tools that help assess the relevance of results. The segmentation and particle-linking steps are separated when

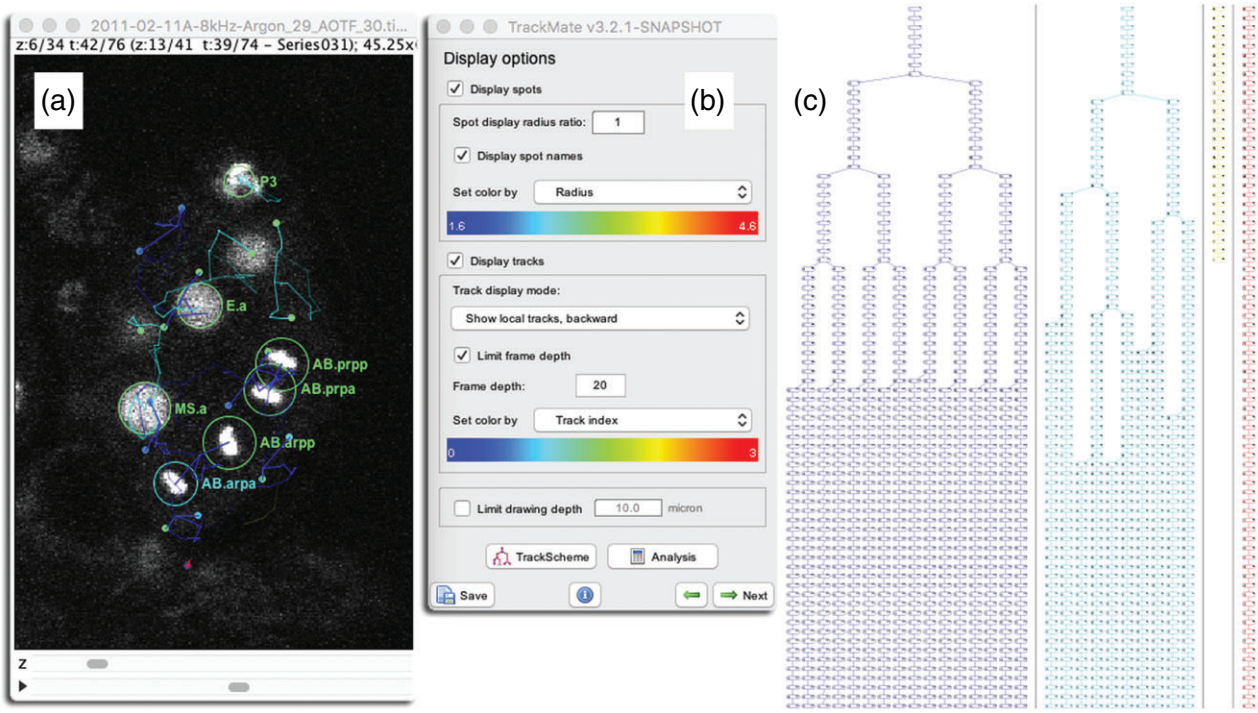

FIGURE 4 | Example of *C. elegans* lineage performed in TrackMate. H2B-GFP *C. elegans* embryos were prepared and imaged on a laser-scanning confocal microscope.[66] TrackMate was used to segment and track *C. elegans* nuclei to semi-automatically generate a full lineage (a, b). The lineage is made of four tracks: the lineage of the AB progenitor, the lineage of the P1 progenitor, and the two basal bodies tracks, followed up to their disappearance (c).

working through the wizard-like graphical user interface (GUI), where at each step the results of the provided algorithms are visualized for user-based assessment and correction. Various tools for segmentation and/or track analysis for numerical results are provided, and plots can be generated directly within the plugin; furthermore, results can be exported to other software such as MATLAB for further analysis.

As there is no single, universal, optimal tracking algorithm that sufficiently meets the challenges posed by the variation of datasets within the life sciences, TrackMate provides a platform where users and developers can contribute their own detection, particle linking, visualization, or analysis modules. TrackMate is a framework that enables researchers to focus on developing new algorithms, removing the burden of also writing a GUI, visualization and analysis tools, and exporting facilities.

SIGNAL QUANTIFICATION

What information does a pixel value hold in an image? As shown above, that information can be used to group pixels—i.e., segmentation—to map pixels of similar coordinates—i.e., registration—or to follow grouped pixels over time—i.e., tracking. However, that pixel information itself, the intensity of a signal, can be translated into a quantifiable dataset. This is a key task in bioimage analysis, and in particular, for signal quantification techniques, including fluorescence lifetime analysis and colocalization, both of which will be discussed in more detail in the following sections.

Lifetime Analysis

The fluorescence lifetime of a given molecule is the average decay time from its excited state;[37] each fluorescent molecule has a unique lifetime signature that can be spatiotemporally mapped within a cell, tissue, or even on a whole-organism scale. Fluorescence lifetime microscopy (FLIM) and spectral lifetime imaging (SLIM)[4] can be used to assess the state of the environment around a molecule, as fluorescence lifetimes are affected by several factors including pH, oxygen, Ca^{2+} concentrations (dynamic/Stern-Volmer quenching), and molecular interactions via Förster resonance energy transfer (FRET).[3] FLIM also provides great insight into the metabolic state of a cell via endogenous fluorophores, such as the metabolites FAD and NADH, which provide optical biomarkers and have been used to examine changing metabolic activity

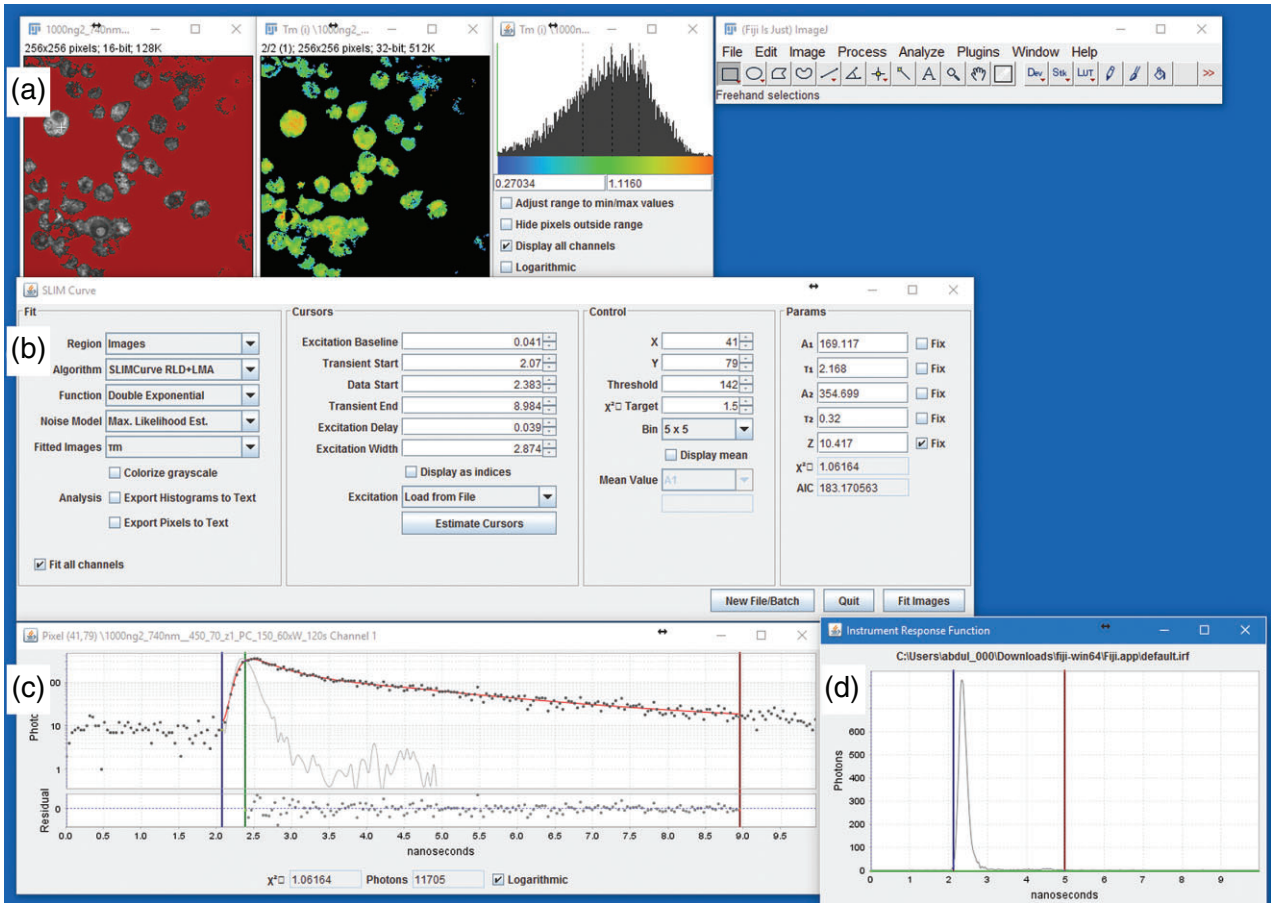

FIGURE 5 | Fluorescence lifetime microscopy (FLIM) images analyzed with spectral lifetime imaging (SLIM) Curve. The figure represents a NADH lifetime of microglia cells activated with lipopolysaccharide (LPS). 740-nm excitation wavelength was used for NADH excitation, with a 450/70 emission filter. FLIM data were acquired with a Becker and Hickl-830 board for 60 seconds on a multiphoton microscope. The analysis was performed using the SLIM Curve plugin for ImageJ (b). This image represents FLIM analysis with a 2-component fit and 5 × 5 binning. The fitted image represents mean lifetime, which is the proportional combination of the free and bound lifetime. The histogram and the color-coded bar represents the distribution of mean lifetime (a). Exponential decay for a single pixel is shown in distribution (c). The Instrument Response Function (IRF) is also shown (d).

within tumor microenvironments.[51] Lifetimes are not influenced by probe concentrations, photobleaching, excitation light intensity, or scattering, making FLIM a direct quantitative measurement and, therefore, a powerfully informative technique that can readily be applied to living cells.

Essentially, when a single molecule or fluorophore is excited, there exists a certain probability that it can return to the ground state, thereby emitting a photon. This temporal decay can be assumed as an exponential decay probability function. There are two ways to measure decay profiles: time domain (photon-counting, TD-FLIM) and frequency domain (FD-FLIM);[18] this review will focus on the former method. For TD-FLIM, a photon distribution histogram can be measured by using time-correlated single photon counting (TCSPC) or fast-gated image intensifiers. Measurements require 'short' (relative to fluorescence lifetime), high-intensity, laser-pulsed excitations and fast detection circuits. TCSPC records photon arrival times at each spatial location after a sufficient number of events have been recorded, building up the histogram over time; conversely, fast-gated image intensifiers measure fluorescence intensity in various time windows across the time range of the fluorescence decay of the sample. In both of these TD-FLIM techniques, because the pulse is not infinitely small, it has its own time profile (instrument response function, IRF) and therefore convolves the decay profile.

SLIM Curve

SLIM Curve is an exponential curve-fitting library used for FLIM and SLIM data analysis (https://slim-

curve.github.io/), as well as an ImageJ plugin that provides the ability to analyze FLIM and SLIM data within ImageJ using the SLIM Curve library. Figure 5 shows a real-world biological example where the NADH lifetime of microglia cells is measured upon activation by lipopolysaccharide (LPS). In addition to this ImageJ plugin, the library itself can be used standalone or via external applications, including MATLAB and Tri2 (FLIM-centric, Windows application). A significant advantage to using the SLIM Curve plugin is full, integrated access with related ImageJ workflows such as segmentation. Both manual and automatic segmentation applications can be used with FLIM analysis. For example, the plugin is compatible with the TWS plugin discussed earlier, aiding in the streamlining of larger workflows.

Colocalization

Colocalization is the measurement of spatial relationships between molecules. Because this is a measurement of codistribution or association of two probes, and not direct interactions, colocalization is best suited to investigate the general locale of a molecule and its potential association with a cellular structure, compartment, or molecular complex. There are two qualitatively differing methods that can be applied to measure these relationships: the 'classical' pixel-based methods that measure global correlation coefficients from pixel intensities for multiple color channels directly, and object-based methods that first segment distinct molecular 'spots' to then analyze their relative spatial distributions; all techniques have been thoroughly reviewed.[6,15,35,36] Pixel-based methods will be the focus of this review.

Several quantitative values are commonly produced in colocalization analyses. Pearson's correlation coefficient (PCC) is a good estimate of overall association between probes, as it measures pixel-by-pixel correlation, mean-normalized to values from −1 (anticorrelation) to 1 (correlation).[38] Manders' colocalization coefficient (MCC) measures the fraction of total probe signal that colocalizes with another signal independent of proportionality.[39] To identify the background, the Costes approach chooses a threshold so that the PCC calculated from pixel intensities below the threshold is zero or negative, and Costes' randomization reveals the significance of calculated Pearson's and Manders' coefficients by iteratively generating 'random' scrambled blocks of pixels to directly compare with the unscrambled image.[16]

As any signal is considered 'real' in pixel-based methods, an overestimation of colocalization can occur. Careful emission spectra and filter selection should be used, as well as sequential imaging, to avoid false colocalization signal. In order to counteract these effects, one must carefully and properly preprocess their acquired data, applying noise-removing filters, thresholds, multiple Gaussian fits, and/or limiting search space with regions of interest (ROIs). Finally, zero–zero pixels are not biologically relevant and should be excluded if possible to avoid skewing colocalization statistics.

Coloc 2

Coloc 2 is a colocalization tool available in ImageJ, performing pixel-based intensity correlation analyses (http://imagej.net/Coloc_2). Figure 6 reveals the Coloc 2 user interface, which displays images for colocalization studies of HIV maturation, examining the spatial location of HIV proteins within certain host cell compartments. The current package of Coloc 2 excludes object-based overlap analysis. As it stands, Coloc 2 requires at least two single channel images opened and displayed in ImageJ; currently, *z*-stacks can be processed, but not time series. ROIs or binary masks can be applied to restrict the region used in the analysis, which is useful for eliminating zero–zero regions, but care should be taken to avoid introducing bias. The user can individually select their preferred algorithms and calculated statistics. Datasets can be exported as PDFs, which include images of generated scatter plots, 2D intensity histograms, individual channel displays, as well as the full list of calculated data values to help facilitate comparisons between different colocalization experiments. Coloc 2 also uses ImgLib2 to implement all methods above independent of pixel type (e.g., 8-, 16-, 32-bit, and more), enabling efficient extensibility and compatibility within the ImageJ ecosystem.

EVOLVING TOOLS IN IMAGEJ

It is also important to highlight particular techniques in image analysis that would benefit from further exposure and development of tools within ImageJ. Due to the continual growth of bioimaging in terms of advancing instrumentation and continued data complexity, these image techniques will be necessary to meet the needs of researchers now and in the near future. Two in particular that have been well-vetted tools by the microscopy community, though still evolving in ImageJ, are deconvolution and spectral analysis that we review here.

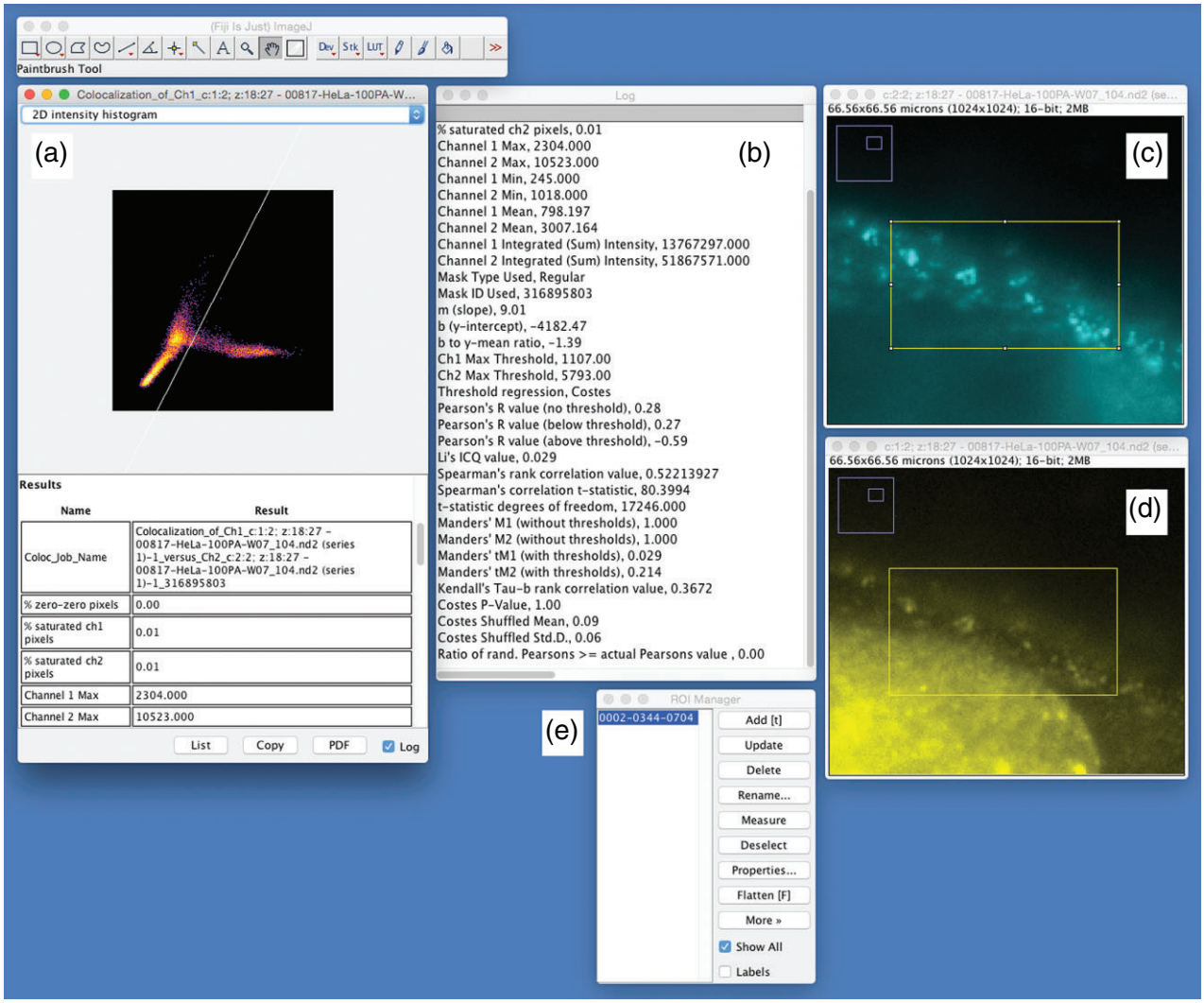

FIGURE 6 | An ImageJ plugin for colocalization, Coloc 2. HeLa cells were transfected with HIV-1 Gag-CFP (c) and RNA-tagging protein (MS2-YFP-NLS) (d), fixed at 24-h posttransfection, and imaged with 100× Plan Apo (NA = 1.45). After applying a region of interest (e), the Coloc 2 output (a) can be exported as PDF and is also summarized in the log window (b). Images are courtesy of Jordan Becker from the laboratory of Nathan Sherer at the University of Wisconsin–Madison.

Deconvolution

Deconvolution is one of the most common image reconstruction tasks that arise in 3D fluorescence microscopy. Deconvolution is essentially an algorithm-based process of reversing distortions on a resultant image created by an optical system (i.e., a fluorescence microscope); it is a mathematical operation to restore an image that is degraded by a physical process, a convolution. Every optical microscope, no matter the technique or method employed, is an instrument that convolves reality; it is the nature of optics. Therefore, any resulting image, in order to return to a true representation of the original object, must be mathematically deconvolved.

An acquired image arises from the convolution of the light source, the original object, and a PSF. The PSF of an optical device is the image of a single, subresolution point object. The PSF is the smallest unit that builds an acquired image in its entirety, and an image is computed as the sum of all point objects. A convolution is replacing each light source by its corresponding PSF to produce a 'blurry' image; therefore, a deconvolution is the reverse process, collecting all the 'blurry' light and putting it back to its original source location. An image from a fluorescence microscope is completely described by its PSF; therefore, knowledge of a system's PSF is an essential step in deconvolution. There exist multiple ways to

determine a PSF. A theoretical PSF can be computed given the model of the microscope and known microscopic parameters, including numerical aperture, excitation wavelength, refractive indices and mounting medium, etc. An experimental PSF is more accurate and can be acquired directly on the same microscopic system used for image acquisitions via imaging of microscopic spherical beads.

In general, deconvolution results in better looking images, improved identification of features, removal of artifacts (due to PSF calibration), higher resolutions, and, last but not least, better quantitative image analyses (reviewed in Ref 65). Instead of discarding the out-of-focus signal, it is reassigned to the correct image plane. Thus, the blurred signal is moved back into focus, which increases definition of the object by improving the signal-to-noise ratio and contrast.

Many users are unaware that deconvolution techniques are 'hidden' in other tools available in ImageJ. For example, within the Multiview Reconstruction plugin, specific for multiview datasets, there are accessible deconvolution techniques that can be easily applied to any 3D$^+$ dataset and are pushing boundaries in efficient implementation of deconvolution algorithms for multiview images.[50,58] While a lot of effort has gone into deconvolution approaches in ImageJ in recent years, there is currently no single flagship deconvolution plugin available as part of the open source ImageJ ecosystem. Instead, the ImageJ development team is making a concerted effort to include deconvolution algorithms as part of the core ImageJ Ops library for image analysis (https://imagej.github.io/presentations/2015-09-04-imagej2-deconvolution/). Ops provides unified interfaces for implementations of image-processing algorithms[57] (http://imagej.net/ImageJ_Ops). As ImageJ Ops matures, its built-in functionality will become increasingly accessible to end users in the form of plugins.

Spectral Analysis

There is an ever-growing variety of available fluorophores for biological labeling, which continually adds to the complexities of bioimaging. It is not always possible to combine fluorophores whose emission spectra do not overlap or to have the optimal optical filters to properly separate emission spectra. There is currently a strong interest in spectral unmixing techniques. Spectral unmixing involves the separation of emission spectra from multiple fluorophores postacquisition. By collecting nearly all fluorescence emitted without differentiating individual fluorescent molecules, spectral imaging combined with quantitative unmixing is able to overcome many limitations in fluorescence microscopy. Specific fluorophore emissions can be extracted from total signal and their intensities properly redistributed to restore the true signal, which is then no longer affected by overlapping signals.

This ability to distinguish highly overlapping emission spectra vastly extends the possibilities in multicolor imaging. Spectral unmixing can also be used to remove autofluorescence signal from samples, a very common occurrence in tissue samples, in order to separate 'real' signal from autofluorescence and can allow the distinction between FRET emission and donor bleed through, termed spectral FRET, resulting in more accurate and quantitative FRET analysis. Multichannel fluorescence imaging also allows various aspects of the same specimen to be examined simultaneously.

Spectral imaging eliminates the need for specific filters, since all filtering can be done computationally after the acquisition process; however, there remains a need to transfer existing tools into the biological world. In general, spectral unmixing algorithms have been widely applied and optimized in the field of astronomy. Despite the power of this technique, only a handful of minimalist tools are available within ImageJ. PoissonNMF allows the decomposition of lambda stacks into single spectra of fluorescently labeled samples. It can be used without reference spectra, estimating spectra by using nonnegative matrix factorization, which is suitable for data with high levels of shot-noise. Even without prior knowledge or minimal knowledge of the sample spectra, the data can be decomposed through the application of 'blind unmixing.'[43] The simple matrix algorithms for spectral unmixing used in the plugin, Spectral Unmixing, allow the measurement of the spectral bleed through between color channels from reference images.[42,70] Essentially, the relative intensity of each individual fluorophore is stored in a 'mixing matrix.' The inverse of this matrix is used to correct bleed through seen in experimental images, recorded under the same conditions as the reference images. But again, this is only the beginning of what can be done with this technique, and reveals another avenue for needed development in currently available ImageJ plugins.

CONCLUSION

Open-Source Tools in Image Analysis

Continued advancements in image processing have opened a whole new realm in biological imaging and

data analysis. Current computational tools and bioimage analysis techniques have transformed images from only visual, qualitative observations into robust, quantitative measurements yielding complex, oftentimes multiparametric datasets from which relevant information can be computationally extracted. Through collaborations between biologists and computer scientists over the past decade, ImageJ has rapidly emerged as the cornerstone for such progressive, joint endeavors. The need for reproducibility makes open-source methods paramount for scientific transparency. However, open-source tools are an often-misunderstood element. These community-based, collaborative applications, especially in the case of ImageJ, are constantly evolving and are by no means a final product, but are projects that constantly undergo updates and development, perhaps much to the occasional annoyance of the user-base. As well, they are not anticommercial; in fact, many open-source projects enjoy the direct contribution of commercial entities. For example, in the world of image databases, the Open Microscopy environment (OME)[1] enjoys the strong participation of a driving member, Glencoe Inc. The Micro-Manager software package[19] for acquisition is led by Open Imaging, Inc. Many more companies utilize and contribute. It is essential to realize that the progress on the computational side of open-source image analysis is as evolving and nonstagnant as the biology itself; as in biological discovery, these computational foundations are continuously being built upon.

Building New Bridges in Open Source: Statistics

ImageJ is capable of fluid interconnectivity with an extensive network of other open-source applications. Scientific research is not a standalone endeavor; interoperability and collaboration lead to innovation and discovery. While there is always a continued need for communication between computer scientists and biologists to keep the field of bioimage analysis moving forward, we must also consider what other opportunities exist for mutual learning and exchange. In particular, we believe the near future of ImageJ development should include creating effective bridges for statisticians to bring novel algorithms to bioimage analysis via ImageJ. This is necessary for continued progress in the field,[10] and with the increasing complexity of quantitative data extraction, it is more important than ever for algorithms to be implemented in a statistically robust way. In particular, it is becoming clear that more principled approaches are needed in terms of, for example, stochastic modeling and uncertainty quantification, to appropriately account for 'noises' of various different nature, especially for high throughput imaging applications. ImageJ provides an accessible avenue for such methods to be applied by developers and biologists to reveal the statistical relevance of their datasets. There is already some promising work in this area to provide bridge tools that statisticians use, such as R (https://www.r-project.org/) and Bioconductor,[23] and we expect this to also grow within ImageJ as these statistical approaches are explored and adapted for biological imaging.

Biologists in the World of Computational Imaging

We know ImageJ is not all inclusive in its image analysis support, and as a result, this review is not all encompassing. New approaches and tools are continually being added through user requests and direct contributions. As a result, we have focused on exposing both new and experienced ImageJ users to various existing and emerging applications and techniques, as opposed to providing detailed 'how-to' instructions. Through this exposure to specific functionality, we hope to provide insight into tools that exist to address specific biological problems, and more importantly, to reveal an entire community of support. We hope this review provides a launch pad for further open source and ImageJ exploration.

The most important aspect of open-source image processing is not the individual algorithm or process, but rather the biologists, developers, and statisticians that drive the development, application, and maintenance of a given method. In a review such as this that aims to make ImageJ, and open-source tools in general, more accessible to the bench biologist, the most important message that can be sent to existing and prospective users is to be active and not silent in one's use, development, application, and implementation of a process—whether ultimately successful or not.

At this point, collaboration is simply a matter of revealing new opportunities. If you develop an analysis workflow or create a tool or algorithm for use in your local environment that could be useful to the greater community, as was the case for many of the plugins discussed in this review, consider distributing it as an ImageJ plugin. Communication is key to continuing collaborations, which require exposure in a public venue, the most accessible being the ImageJ Forum. Reporting of one's efforts at any level, we have found, always has utility to the community. Not only does such communication and sharing

promote standardization, reducing costs and resource requirements,[5] but just as importantly, it introduces us to new communities, opens up new collaborations, and results in improved analysis approaches, unveiling avenues for computational improvements and developments, which ultimately benefits all.

ACKNOWLEDGMENTS

We acknowledge manuscript edits and figure contributions from Nathan Sherer and Jordan Becker, Molecular Virology and Oncology, University of Wisconsin–Madison; Alison Walter, LOCI University of Wisconsin-Madison, MD Adbul Sagar, LOCI, University of Wisconsin–Madison; and Cole Drifka, LOCI, University of Wisconsin–Madison.

We also acknowledge contributions from members of the ImageJ community: Ignacio Arganda-Carreras, TWS, University of the Basque Country; Aivar Grislis and Abdul Kader Sagar, SLIM Curve, LOCI, University of Wisconsin–Madison; Paul Barber, SLIM Curve, CRUK/MRC, University of Oxford; Stephan Preibisch, Multiview Reconstruction, MPI-CBG; Albert Cardona and Stephan Saalfeld, TrakEM2, HHMI Janelia Research Campus; Tobias Pietzsch, BigDataViewer, MPI-CBG; Florian Jug, ClearVolume, MPI-CBG; Jean-Yves Tinevez, TrackMate, Imagopole, Institut Pasteur; Dan White, Coloc 2, MPI-CBG; Tom Kazimiers, Coloc 2; Brian Northan, True North Intelligent Algorithms LLC; Johannes Schindelin, Fiji core and plugins, MPI-CBG and LOCI. This list is by no means exhaustive, and therefore we wish to also thank the broader ImageJ community and contributions of all its members.

Funding sources include Morgridge Interdisciplinary Fellows Program (ETA) and funding from the Morgridge Institute for Research (KWE) and the UW Laboratory for Optical and Computational Instrumentation (KWE).

REFERENCES

1. Allan C, Burel J-M, Moore J, Blackburn C, Linkert M, Loynton S, Macdonald D, Moore WJ, Neves C, Patterson A, et al. OMERO: flexible, model-driven data management for experimental biology. *Nat Methods* 2012, 9:245–253. doi:10.1038/nmeth.1896.
2. Arganda-Carreras I, Kaynig V, Schindelin J, Cardona A, Seung HS. *Trainable Weka Segmentation: A Machine Learning Tool for Microscopy Image Segmentation*. 2014, 73–80. Available at: http://doi.org/10.5281/zenodo.59290.
3. Becker W. The TCSPC handbook. *Scanning* (6th edition). Berlin, Germany: Becker and Hickl GmbH; 2010, 1–566.
4. Bird DK, Eliceiri KW, Fan C-H, White JG. Simultaneous two-photon spectral and lifetime fluorescence microscopy. *Appl Opt* 2004, 43:5173–5182. doi:10.1364/AO.43.005173.
5. Blind K, Mangelsdorf A, Jungmittag A. *The Economic Benefits of Standardization: An Update of the Study Carried Out by DIN in 2000*. Berlin, Germany: DIN German Institute for Standardization; 2011, 1–20.
6. Bolte S, Cordelieres FP. A guided tour into subcellular colocalisation analysis in light microscopy. *J Microsc* 2006, 224:13–232. doi:10.1111/j.1365-2818.2006.01706.x.
7. Bonneau S, Dahan M, Cohen LD. Single quantum dot tracking based on perceptual grouping using minimal paths in a spatiotemporal volume. *IEEE Trans Image Process* 2005, 14:1384–1395. doi:10.1109/TIP.2005.852794.
8. Cardona A, Hartenstein V, Saalfeld S, Preibisch S, Schmid B, Cheng A, Pulokas J, Tomancak P. An integrated micro- and macroarchitectural analysis of the Drosophila brain by computer-assisted serial section electron microscopy. *PLoS Biol* 2010, 10:8. doi:10.1371/journal.pbio.1000502.
9. Cardona A, Saalfeld S, Schindelin J, Arganda-Carreras I, Preibisch S, Longair M, Tomancak P, Hartenstein V, Douglas RJ. TrakEM2 software for neural circuit reconstruction. *PLoS One* 2012, 7:e38011. doi:10.1371/journal.pone.0038011.
10. Cardona A, Tomancak P. Current challenges in open-source bioimage informatics. *Nat Methods* 2012, 9:661–665. doi:10.1038/nmeth.2082.
11. Carpenter AE, Kamentsky L, Eliceiri KW. A call for bioimaging software usability. *Nat Methods* 2012, 9:666–670. doi:10.1038/nmeth.2073.
12. Chen Y, Wang Y, Zhang J, Deng Y, Jiang L, Song E, Wu XS, Hammer JA, Xu T, Lippincott-Schwartz J. Rab10 and myosin-va mediate insulin-stimulated GLUT4 storage vesicle translocation in adipocytes. *J Cell Biol* 2012, 198:545–560. doi:10.1083/jcb.201111091.

13. Chenouard N, Smal I, de Chaumont F, Maška M, Sbalzarini IF, Gong Y, Cardinale J, Carthel C, Coraluppi S, Winter M, et al. Objective comparison of particle tracking methods. *Nat Methods* 2014, 11:281–289. doi:10.1038/nmeth.2808.

14. Christensen RP, Bokinsky A, Santella A, Wu Y, Marquina-Solis J, Guo M, Kovacevic I, Kumar A, Winter PW, Tashakkori N, et al. Untwisting the *Caenorhabditis elegans* embryo. *Elife* 2015, 4:e10070. doi:10.7554/eLife.10070.

15. Comeau JWD, Costantino S, Wiseman PW. A guide to accurate fluorescence microscopy colocalization measurements. *Biophys J* 2006, 91:4611–4622. doi:10.1529/biophysj.106.089441.

16. Costes SV, Daelemans D, Cho EH, Dobbin Z, Pavlakis G, Lockett S. Automatic and quantitative measurement of protein-protein colocalization in live cells. *Biophys J* 2004, 86:3993–4003. doi:10.1529/biophysj.103.038422.

17. Damiano L, Stewart KM, Cohet N, Mouw JK, Lakins JN, Debnath J, Reisman D, Nickerson JA, Imbalzano AN, Weaver VM. Oncogenic targeting of BRM drives malignancy through C/EBPβ-dependent induction of α5 integrin. *Oncogene* 2014, 33:2441–2453. doi:10.1038/onc.2013.220.

18. Digman MA, Caiolfa VR, Zamai M, Gratton E. The phasor approach to fluorescence lifetime imaging analysis. *Biophys J* 2008, 94:L14–L16. doi:10.1529/biophysj.107.120154.

19. Edelstein AD, Tsuchida MA, Amodaj N, Pinkard H, Vale RD, Stuurman N. Advanced methods of microscope control using μManager software. *J Biol Methods* 2014, 1:e10. doi:10.14440/jbm.2014.36.

20. Eliceiri KW, Berthold MR, Goldberg IG, Ibáñez L, Manjunath BS, Martone ME, Murphy RF, Peng H, Plant AL, Roysam B, et al. Biological imaging software tools. *Nat Methods* 2012, 9:697–710. doi:10.1038/nmeth.2084.

21. Entenberg D, Rodriguez-Tirado C, Kato Y, Kitamura T, Pollard JW, Condeelis J. In vivo subcellular resolution optical imaging in the lung reveals early metastatic proliferation and motility. *Intravital* 2015, 4:1–11. doi:10.1080/21659087.2015.1086613.

22. Genovesio A, Liedl T, Emiliani V, Parak WJ, Coppey-Moisan M, Olivo-Marin JC. Multiple particle tracking in 3-D + t microscopy: method and application to the tracking of endocytosed quantum dots. *IEEE Trans Image Process* 2006, 15:1062–1070. doi:10.1109/TIP.2006.872323.

23. Gentleman RC, Carey VJ, Bates DM, Bolstad B, Dettling M, Dudoit S, Ellis B, Gautier L, Ge Y, Gentry J, et al. Bioconductor: open software development for computational biology and bioinformatics. *Genome Biol* 2004, 5:R80. doi:10.1186/gb-2004-5-10-r80.

24. Gerhold AR, Ryan J, Vallée-Trudeau JN, Dorn JF, Labbé JC, Maddox PS. Investigating the regulation of stem and progenitor cell mitotic progression by in situ imaging. *Curr Biol* 2015, 25:1123–1134. doi:10.1016/j.cub.2015.02.054.

25. Girstmair J, Zakrzewski A, Lapraz F, Handberg-Thorsager M, Tomancak P, Pitrone PG, Simpson F, Telford MJ. Light-sheet microscopy for everyone? Experience of building an OpenSPIM to study flatworm development. *BMC Dev Biol* 2016, 16:22. doi:10.1186/s12861-016-0122-0.

26. Godinez WJ, Lampe M, Wörz S, Müller B, Eils R, Rohr K. Deterministic and probabilistic approaches for tracking virus particles in time-lapse fluorescence microscopy image sequences. *Med Image Anal* 2009, 13:325–342. doi:10.1016/j.media.2008.12.004.

27. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The WEKA data mining software. *SIGKDD Explor Newsl* 2009, 11:10. doi:10.1145/1656274.1656278.

28. Han H-S, Niemeyer E, Huang Y, Kamoun WS, Martin JD, Bhaumik J, Chen Y, Roberge S, Cui J, Martin MR, et al. Quantum dot/antibody conjugates for in vivo cytometric imaging in mice. *Proc Natl Acad Sci USA* 2015, 112:1–6. doi:10.1073/pnas.1421632111.

29. Huisken J, Swoger J, Bene FD, Wittbrodt J, Stelzer EHK. Optical sectioning deep inside live embryos by selective plane illumination microscopy. *Science* 2004, 305:1007–1009.

30. Icha J, Schmied C, Sidhaye J, Tomancak P, Preibisch S, Norden C. Using light sheet fluorescence microscopy to image zebrafish eye development. *J Vis Exp* 2016, 110:e53966. doi:10.3791/53966.

31. Jaqaman K, Loerke D, Mettlen M, Kuwata H, Grinstein S, Schmid SL, Danuser G. Robust single-particle tracking in live-cell time-lapse sequences. *Nat Methods* 2008, 5:695–702. doi:10.1038/nmeth.1237.

32. Jug F, Pietzsch T, Kainm D, Funke J, Kaiser M, Nimwegen EV, Rother C, Myers G. Optimal joint segmentation and tracking of *Escherichia coli* in the mother machine. In: Cardoso MJ, Simpson I, Arbel T, Precup D, Ribbens A, eds. *Bayesian and Graphical Models for Biomedical Imaging*. Switzerland:Springer International Publishing; 2014, 25–36. doi:10.1007/978-3-319-12289-2.

33. Jug F, Pietzsch T, Preibisch S, Tomancak P. Bioimage informatics in the context of Drosophila research. *Methods* 2014b, 68:60–73. doi:10.1016/j.ymeth.2014.04.004.

34. Keller PJ, Schmidt AD, Wittbrodt J, Stelzer EHK. Reconstruction of zebrafish early embryonic development by scanned light sheet microscopy. *Science* 2008, 322:1065–1069. doi:10.1126/science.1162493.

35. Kopan R, Goate A, Dunn KW, Kamocka MM, Mcdonald JH. A common enzyme connects notch

signaling and Alzheimer's disease. *AJP Cell Physiol* 2011, 46202:723–742. doi:10.1152/ajpcell.00462.2010.

36. Lagache T, Sauvonnet N, Danglot L, Olivo-Marin JC. Statistical analysis of molecule colocalization in bioimaging. *Cytometry A* 2015, 87:568–579. doi:10.1002/cyto.a.22629.
37. Lakowicz JR, Szmacinski H, Nowaczyk K, Berndt KW, Johnson M. Fluorescence lifetime imaging. *Anal Biochem* 1992, 202:316–330. doi:10.1016/0003-2697(92)90112-K.
38. Manders EMM, Stap J, Brakenhoff GJ, Van Driel R, Aten A. Dynamics of three-dimensional replication patterns during the S-phase, analysed by double labelling of DNA and confocal microscopy. *J Cell Sci* 1992, 103(pt 3):857–862.
39. Manders EMM, Verbeek FJ, Ate JA. Measurement of co-localisation of objects in dual-colour confocal images. *J Microsc* 1993, 169:375–382. doi:10.1111/j.1365-2818.1993.tb03313.x.
40. Meijering E. Cell segmentation: 50 years down the road. *IEEE Signal Process Mag* 2012, 29:140–145. doi:10.1109/MSP.2012.2204190.
41. Meyer AS, Hughes-Alford SK, Kay JE, Castillo A, Wells A, Gertler FB, Lauffenburger DA. 2D protrusion but not motility predicts growth factor-induced cancer cell migration in 3D collagen. *J Cell Biol* 2012, 197:721–729. doi:10.1083/jcb.201201003.
42. Neher R, Neher E. Optimizing imaging parameters for the separation of multiple labels in a fluorescence image. *J Microsc* 2004, 213:46–62. doi:10.1111/j.1365-2818.2004.01262.x.
43. Neher RA, Mitkovski M, Kirchhoff F, Neher E, Theis FJ, Zeug A. Blind source separation techniques for the decomposition of multiply labeled fluorescence images. *Biophys J* 2009, 96:3791–3800. doi:10.1016/j.bpj.2008.10.068.
44. Paguirigan AL, Beebe DJ. Microfluidics meet cell biology: bridging the gap by validation and application of microscale techniques for cell biological assays. *Bioessays* 2008, 30:811–821. doi:10.1002/bies.20804.
45. Peng H. Bioimage informatics: a new area of engineering biology. *Bioinformatics* 2008, 24:1827–1836. doi:10.1093/bioinformatics/btn346.
46. Pietzsch T, Preibisch S, Tomancák P, Saalfeld S. ImgLib2—generic image processing in Java. *Bioinformatics* 2012, 28:3009–3011. doi:10.1093/bioinformatics/bts543.
47. Pietzsch T, Saalfeld S, Preibisch S, Tomancak P. BigDataViewer: visualization and processing for large image data sets. *Nat Methods* 2015, 12:481–483. doi:10.1038/nmeth.3392.
48. Polan DF, Brady SL, Kaufman RA. Tissue segmentation of computed tomography images using a Random Forest algorithm: a feasibility study. *Phys Med Biol* 2016, 61:6553–6569. doi:10.1088/0031-9155/61/17/6553.
49. Preibisch S, Amat F, Stamataki E, Sarov M, Singer RH, Myers E, Tomancak P. Efficient Bayesian-based multiview deconvolution. *Nat Methods* 2014, 11:645–648. doi:10.1038/nmeth.2929.
50. Preibisch S, Saalfeld S, Schindelin J, Tomancak P. Software for bead-based registration of selective plane illumination microscopy data. *Nat Methods* 2010, 7:418–419. doi:10.1038/nmeth0610-418.
51. Provenzano PP, Eliceiri KW, Keely PJ. Multiphoton microscopy and fluorescence lifetime imaging microscopy (FLIM) to monitor metastasis and the tumor microenvironment. *Clin Exp Metastasis* 2009, 26:357–370. doi:10.1007/s10585-008-9204-0.
52. Royer LA, Weigert M, Günther U, Maghelli N, Jug F, Sbalzarini IF, Myers EW. ClearVolume: open-source live 3D visualization for light-sheet microscopy. *Nat Methods* 2015, 12:480–481. doi:10.1038/nmeth.3372.
53. Russell MR, Lerner TR, Burden JJ, Nkwe DO, Pelchen-Matthews A, Domart MC, Durgan J, Weston A, Jones ML, Peddie CJ, et al. 3D correlative light and electron microscopy of cultured cells using serial blockface scanning electron microscopy. *J Cell Sci* 2016. doi:10.1242/jcs.188433.
54. Saalfeld S, Cardona A, Hartenstein V, Tomancak P. As-rigid-as-possible mosaicking and serial section registration of large ssTEM datasets. *Bioinformatics* 2010, 26:i57–i63. doi:10.1093/bioinformatics/btq219.
55. Sbalzarini IF, Koumoutsakos P. Feature point tracking and trajectory analysis for video imaging in cell biology. *J Struct Biol* 2005, 151:182–195. doi:10.1016/j.jsb.2005.06.002.
56. Schindelin J, Arganda-Carreras I, Frise E, Kaynig V, Longair M, Pietzsch T, Preibisch S, Rueden C, Saalfeld S, Schmid B, et al. Fiji: an open-source platform for biological-image analysis. *Nat Methods* 2012, 9:676–682. doi:10.1038/nmeth.2019.
57. Schindelin J, Rueden CT, Hiner MC, Eliceiri KW. The ImageJ ecosystem: an open platform for biomedical image analysis. *Mol Reprod Dev*, 2015, 82:518–529. doi:10.1002/mrd.22489.
58. Schmid B, Huisken J. Real-time multi-view deconvolution. *Bioinformatics* 2015, 31:3398–3400. doi:10.1093/bioinformatics/btv387.
59. Schmid B, Schindelin J, Cardona A, Longair M, Heisenberg M. A high-level 3D visualization API for Java and ImageJ. *BMC Bioinformatics* 2010, 11:274. doi:10.1186/1471-2105-11-274.
60. Schneider CA, Rasband WS, Eliceiri KW. NIH Image to ImageJ: 25 years of image analysis. *Nat Methods* 2012, 9:671–675. doi:10.1038/nmeth.2089.
61. Shah AT, Diggins KE, Walsh AJ, Irish JM, Skala MC. In vivo autofluorescence imaging of tumor

heterogeneity in response to treatment. *Neoplasia* 2015, 17:862–870. doi:10.1016/j.neo.2015.11.006.

62. Shao Z, Watanabe S, Christensen R, Jorgensen EM, Colón-Ramos DA. Synapse location during growth depends on glia location. *Cell* 2013, 154:337–350. doi:10.1016/j.cell.2013.06.028.
63. Smal I, Meijering E, Draegestein K, Galjart N, Grigoriev I, Akhmanova A, van Royen ME, Houtsmuller AB, Niessen W. Multiple object tracking in molecular bioimaging by Rao-Blackwellized marginal particle filtering. *Med Image Anal* 2008, 12:764–777. doi:10.1016/j.media.2008.03.004.
64. Sternberg SR. Biomedical image processing. *Computer* 1983, 16:22–34. doi:10.1109/MC.1983.1654163.
65. Swedlow JR. Quantitative fluorescence microscopy and image deconvolution. In: Sluder G, Wolf DE, eds. *Methods in Cell Biology*. 4th ed. Elsevier Inc; 2013, 407–426. doi:10.1016/B978-0-12-407761-4.00017-8.
66. Tinevez J-Y, Dragavon J, Baba-Aissa L, Roux P, Perret E, Canivet A, Galy V, Shorte S. A quantitative method for measuring phototoxicity of a live cell imaging microscope. *Methods Enzymol* 2012, 506:291–309. doi:10.1016/B978-0-12-391856-7.00039-1.
67. Versaevel M, Grevesse T, Gabriele S. Spatial coordination between cell and nuclear shape within micropatterned endothelial cells. *Nat Commun* 2012, 3:671. doi:10.1038/ncomms1668.
68. Yu HH, Kao CF, He Y, Ding P, Kao JC, Lee T. A complete developmental sequence of a Drosophila neuronal lineage as revealed by twin-spot MARCM. *PLoS Biol* 2010, 8:39–40. doi:10.1371/journal.pbio.1000461.
69. Zhang Y, Legay S, Barrieìre Y, Méchin V, Legland D. Color quantification of stained maize stem section describes lignin spatial distribution within the whole stem. *J Agric Food Chem* 2013, 61:3186–3192. doi:10.1021/jf400912s.
70. Zimmermann T, Rietdorf J, Pepperkok R. Spectral imaging and its applications in live cell microscopy. *FEBS Lett* 2003, 546:87–92. doi:10.1016/S0014-5793(03)00521-0.

