Journal of Neuroscience Methods 324 (2019) 108286

Contents lists available at ScienceDirect

Journal of Neuroscience Methods

journal homepage: www.elsevier.com/locate/jneumeth

A versatile macro-based neurohistological image analysis suite for ImageJ focused on automated and standardized user interaction and reproducible data output

Miky Timothy[a,*], Paul M. Forlano[a,b,c,d,e]

[a] Department of Biology, Brooklyn College, City University of New York, Brooklyn, NY, United States
[b] Doctoral Subprogram in Ecology, Evolutionary Biology and Behavior, The Graduate Center, City University of New York, New York, NY, United States
[c] Doctoral Subprogram in Neuroscience, The Graduate Center, City University of New York, New York, NY, United States
[d] Doctoral Subprogram in Behavioral and Cognitive Neuroscience, The Graduate Center, City University of New York, New York, NY, United States
[e] Aquatic Research and Environmental Assessment Center (AREAC), Brooklyn College, Brooklyn, NY, United States

ARTICLE INFO

Keywords:
Neurohistology
ImageJ
Image analysis
Macro programming
Quantitative neuroanatomy
Immunofluorescence microscopy

ABSTRACT

Background: The development and increasing adoption of advanced microscopy imaging technologies, including high resolution, multi-dimensional digital photography and multiple fluorescence channel acquisition, as well as the availability of inexpensive terabyte-capacity storage, have enabled research laboratories to pursue neurohistological imaging experiments involving multiple neurochemical probes and experimental conditions covering a variety of brain regions. Analyzing and processing the resulting datasets, composed of hundreds of micrographs, presents challenges in ensuring accuracy and reproducibility under demanding time and training constraints.

New method: The 'Custom Macros' plugin suite for ImageJ automates and systematizes user interaction in neurohistological image analysis tasks, including region selection and thresholding, point/object counts, area measurement, batch filter processing, and data review. Written in the accessible ImageJ macro language, the plugin implements a user login-based data storage framework and facilitates inter-laboratory collaboration over cloud file server clients.

Results: A macro-based interface approach integrates dozens of novel operations, software interactions, algorithm calls, and background tasks into individual shortcut commands. Every completed procedure generates image, region, and calibrated measurement records that are saved in a standardized folder structure.

Comparisons with existing methods: Plugin installation adds startup access to a persistent interface layer of extensive and streamlined functionality that is generalizable to a variety of neurohistological contexts, thus providing an efficient and reliable alternative to the use of analysis software in an unstructured, provisional manner that necessitates repeated menu and plugin interaction.

Conclusions: Our free/open-source software provides researchers a straightforward solution to addressing daunting usability and data oversight issues, ultimately making efficient, accessible, and reproducible image analysis methodology attainable for many laboratories.

1. Introduction

Technological advances in microscopy, fluorescence histochemistry, microprocessor speed, and digital storage have made possible the design of neuroscience experiments composed of hundreds of individual neurohistological section micrographs from different subjects, labeled with multiple biochemical markers, and representing numerous experimental conditions (Wallace et al., 2015). Quantitative analyses of these data sets necessitate the use of digital software solutions in the form of specialized scientific applications. ImageJ is an open source image processing and analysis platform developed and funded by the NIH (Schneider et al., 2012). The ImageJ software ecosystem is an unequivocal success story of publicly funded, open, collaborative software development, boasting a diverse and active user community (Schindelin et al., 2015). ImageJ and its numerous free, user-made extensions arguably match or outperform the proprietary solutions

* Corresponding author.
E-mail address: mikytimothy@gmail.com (M. Timothy).

https://doi.org/10.1016/j.jneumeth.2019.04.009
Received 20 October 2018; Received in revised form 16 April 2019; Accepted 18 April 2019
Available online 04 May 2019

currently available for neuroscience research. ImageJ has also been extended with the FIJI software suite, as well as several specialized implementations (Doube et al., 2010; Schindelin et al., 2012).

One of the compelling features of ImageJ is its macro functionality. A macro is a series of interface commands, typically recorded in real-time by users, which can be saved as a text file and run to automate repetitive tasks. These simple macros are interoperable with the ImageJ macro language, a robust scripting language with an extensive built-in function library that can be run and debugged within ImageJ. The ImageJ macro environment and scripting language enable the development of 'plugins' composed of dozens of interconnected macros. In such plugins, macros essentially become software commands.

Quantitative analyses of micrograph images and accompanying data processing and entry tasks are typically performed by observers blind to experimental conditions. Standard analysis goals include anatomical region-of-interest selection, grid-based sub-sampling, signal thresholding and feature segmentation, area measurement, point and object counts, and colocalization (Geuna and Herrera-Rincon, 2015; Jensen, 2013). The aforementioned quantity and complexity of modern neurohistological data sets presents challenges to the timely analysis of an experiment, especially considering the need for precision, consistency, and standardized data entry. Image analysis is often performed by students with no prior experience using image-science software. The need for multiple observers per experiment and turnover of student volunteers puts a premium on training time, with the necessary priority set on establishing a thorough understanding of measurement criteria over user proficiency with software interfaces or data storage best practices. Thus, software ease of use and thorough documentation is crucial for optimizing the performance of students carrying out image analysis work. Furthermore, automated software generation of detailed and predictably organized visual and textual records of procedures performed during image analysis allows senior laboratory investigators to accurately review ongoing and final measurement data without always being physically present.

Recent trends in publishing emphasize the importance of methodological transparency and reproducibility in research (Ince et al., 2012). The immediate availability and open licensing to fellow researchers of code used in studies facilitates repeatability and ease of collaboration in science. Software that systematically generates and stores all relevant measurement raw data also greatly assists in, and in some cases may even enable, future meta-analyses.

2. Materials and methods

2.1. Macro language background

The ImageJ macro language is a built-in package in the ImageJ distribution that provides a programming language for the development of scripts and plugins without using the ImageJ API (application programming interface). There are several advantages to developing with the macro language: 1) The user does not need to be familiar with object oriented programming (OOP) methods (as is the case with the ImageJ API and Java, in general). 2) The macro language is simple to use and syntactically accessible to the novice programmer, but at the same time powerful enough, in the hands of an intermediate-to-advanced user, to be utilized as a general scripting language. 3) The macro language contains a library of hundreds of built-in functions (see Supp. Fig. 1A-B for a definition and examples of built-in functions). Users can incorporate built-in functions into recorded macros and further structure their code with control statements (i.e., if, else, for, while) and their own user-defined functions. 4) The macro language continues to be popular among researchers and benefits from superb documentation and regular development of its function library and functionality. Despite the technical limitations of the macro language in its access to core ImageJ processes and statements and the availability of alternative scripting methods with access to the API (e.g., Jython or Ruby), we believe that, in practice, the macro language is versatile and powerful enough to meet the needs of most working neuroscientists.

The macro language is an interpreted (non-compiled) procedural programming language whose logical structure and syntax very closely resembles C and other "curly brace languages," such as C++ and Java (see (Scott, 2015) for discussion on programming paradigms and syntax). Thus, the expressiveness and capability of the ImageJ macro language is comparable to other C-style scripting languages, (e.g., AWK (Robbins, 2015)) and scripts written in these languages can often easily be adapted for ImageJ macros (Supp. Fig. 1A-B). While macros generated by the macro recorder are limited to lists of interface commands (Table 2, Supp. Fig. 1C), these commands can be incorporated into interactive and generalizable scripts by using built-in functions and persistent variables. Multiple macros can in turn be combined in a single text file into 'macro sets', providing access to numerous custom procedures via keyboard shortcuts, interface buttons, and menus. This approach is, in essence, the structure of Custom Lab Image Analysis/Processing Macros.

2.2. Software, hardware, and materials used

ImageJ version 1.52j was the current release at the time of writing and used for testing. Since ImageJ runs in Java, it is operating system-portable, thus we ran macros with success using Microsoft Windows (XP, 7, 8, 10), Linux (Mint 18.x [Ubuntu-based], antiX 17.x [Debian-based], PCLinuxOS 2018), and Apple Mac OS X. Schematic drawings and charts were created in LibreOffice Draw. The GNU Image Manipulation Program (GIMP) was used to assemble and post-process the micrograph figures. Color levels in Fig. 3 were adjusted for readability and visual clarity. Gamma (gray midpoint) histogram levels were uniformly raised for each micrograph in Fig. 5A–C, likewise for readability, as well as to better resolve image preprocessing results. However, black point histogram levels were not altered, so as to not exclude any signal in their presentation.

Figure micrographs are of brain tissue from the plainfin midshipman fish (*Porichthys notatus*) and were imaged with an Olympus BX61 epifluorescence microscope using a Hamamatsu ORCA-03 G CCD camera and MetaMorph image acquisition software (Molecular Devices, Sunnyvale, CA).

2.3. Source code, licensing, and documentation

We host updated source code and documentation on the Custom Macros homepage (www.ijmacros.com) as well as on our laboratory website (www.forlanolab.com). Setting up a source code repository with distributed version control (e.g., GitLab, Launchpad), is planned for the future. All software we developed is licensed under The GNU General Public License v3.0 (GPLv3), a free and open source license with the goal of sharing code with other researchers, facilitating community review, and encouraging contribution to development.

2.4. A note on terminology

The full name of our ImageJ plugin suite is 'Custom Lab Image Analysis/Processing Macros'. For brevity, we will use the capitalized phrase 'Custom Macros' throughout this document to mean the same thing. Lowercase 'macros' refer to the individual procedures users run within our plugin suite. For clarity, referenced folder directories are *italicized* and filenames are <u>underlined</u>.

3. Results

Table 1 lists and categorizes the command procedures of Custom Macros and the accompanying results section in which they are described.

3.1. Plugin environment

3.1.1. Documentation

Custom Macros help documentation is accessible both through a button added to the ImageJ interface ('question mark' in Figs. 1B, Supp. 1A), keyboard shortcut 'F1', and is provided as a. pdf document.

3.1.2. Installation

The simplest way to install Custom Macros is to use an ImageJ installation zip archive file that is already pre-packaged with our plugin. Since ImageJ is a portable program, installation in Microsoft Windows, Apple OS, or Linux can be accomplished via extraction of the pre-packaged installation archive to any directory (to which the user has write-permissions). Custom Macros can likewise be added manually to an existing installation of ImageJ by extracting the contents of the plugin zip archive to the */ImageJ/Macros/* folder. This adds the folders */CustomMacros_Macros/* and */CustomMacros_Users/*, which contain all scripts and documentation. The file */Macros/StartupMacros.txt* is also overwritten. Deleting this file and the above two folders completely uninstalls Custom Macros. The standard ImageJ installation can be downloaded from the NIH website (ImageJ1 Download, 2018) or installed via a Linux package manager.

3.1.3. Cloud networking capabilities

Custom Macros does not require Internet access or network connectivity to work, and indeed can be run standalone, with full functionality, off of a USB drive. However, our plugin offers the additional capability for ImageJ installations to connect to and interact with distributed file servers, typically through local folders connected to 'cloud' service clients like Dropbox and Google Drive. See Fig. 1A for a detailed overview of Custom Macros networking functionality.

3.1.4. User login, folder and file output, and data workflow

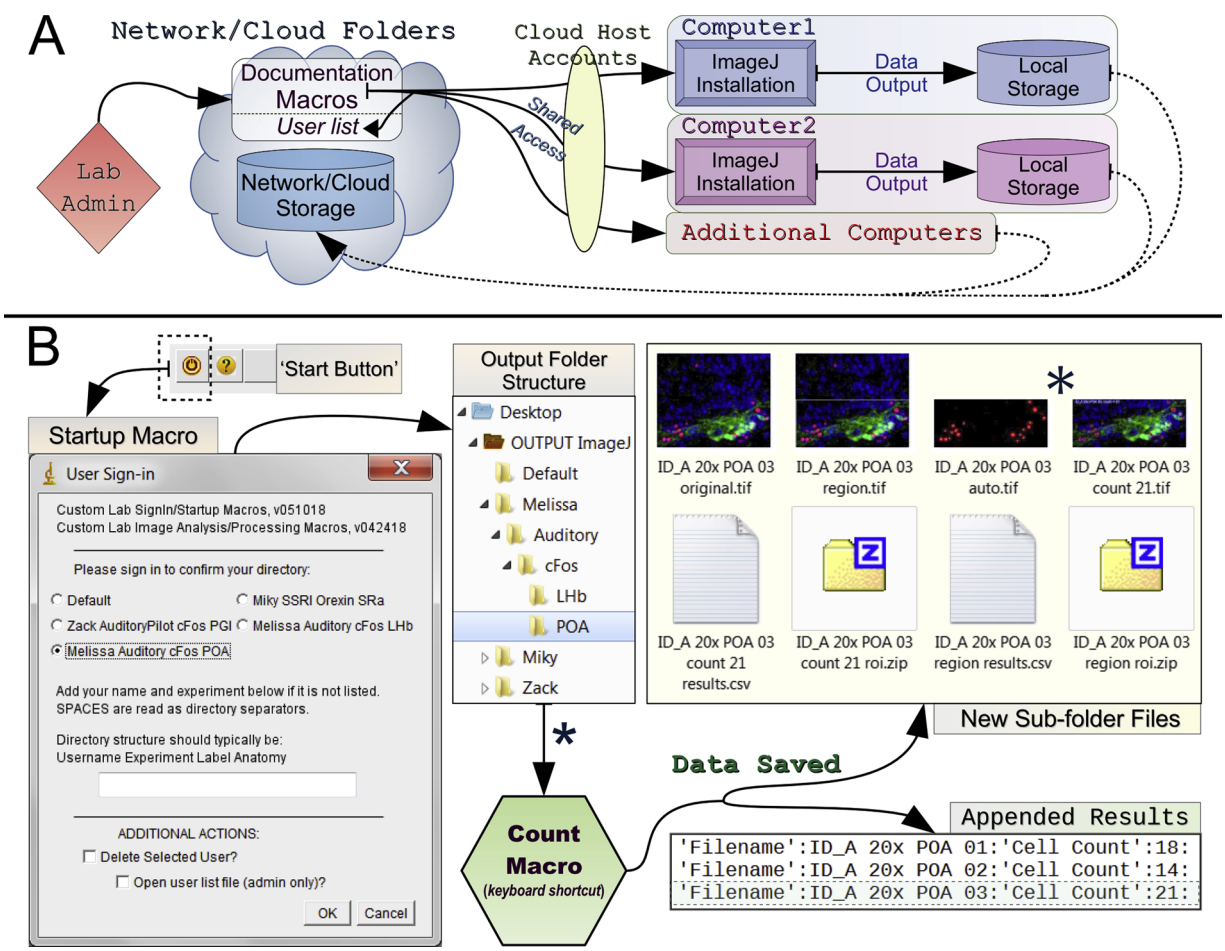
Custom Macros provide a straightforward and standardized workflow environment to ImageJ that facilitates predictably structured data output (Fig. 1B, Supp. Video 1).

3.2. Image preparation

3.2.1. Automated filename-based calibration

Most image analysis tasks generate continuous, spatial measurement data that correspond to pixel values. ImageJ is capable of scaling raw pixel data to real-world units, for example 1 μm^2 per px. Depending on the imaging software source, micrographs may have calibration metadata embedded for this purpose, though they frequently do not, leaving no way for ImageJ to determine how to scale measurements. Moreover, embedded calibration metadata generated by other imaging programs typically cannot be read by ImageJ, though file-import plugins are available that attempt to calibrate images created with other software. Our experience with importing calibration values has been mixed, and this approach introduces additional complexity and uncertainty to an analysis workflow. Proper calibration is crucial to generating accurate data and Custom Macros utilizes a filename-based image scaling approach that is both simple to implement and implicitly reliable.

Our method of filename-based scaling works through the interpretation of 'calibration multiplier values' within image titles. If an image name contains a string of 1–4 integers followed by an x that is bound at the start with a space and at the end with a space, period, or ends the word, that number is read as its calibration multiplier. For



(caption on next page)

Fig. 1. Networking Capabilities and Data Workflow.

A.Networking Capabilities

A lab admin, using a dedicated cloud file host account, creates two folders named */CustomMacros_Macros/ and /CustomMacros_Users/* (abbreviated afterwards for brevity as */_Macros/* and */_users/*, respectively). The admin then uploads the latest plugin files to folder */_Macros/*. The folder */_Users/* is left empty and the system automatically uploads user lists and other user-made data there. Users of ImageJ Custom Macros-equipped computers, with cloud clients installed, request share access to these folders through their laboratory cloud service account. To prevent accidental over-writing of the plugin files, the admin sets the */_Macros/* folder to read-only sharing permissions whereas the */_Users/* folder is made editable by linked accounts. Note that this scheme of a separate lab admin account is best suited for larger laboratories and collaborative efforts; in most cases, the laboratory cloud account can also be the admin account.

Users choose which cloud service or network to use by editing a text file in their local installation folder: */ImageJ/Macros/_Macros/Network.ijm*. This file is simply a pointer to a folder name in the user's home directory. To use Dropbox, the user opens Network.ijm in a text editor, types the word Dropbox, and saves the file. Once connected, the Sign-in/Startup script, upon user initiation (via a 'Start Button' click; Fig. 1B), checks for updated macros (containing bug fixes and added features), documentation, and user-lists in */Dropbox/* and copies these files locally, backing up previous versions. New user sign-ins are saved locally and to */Dropbox/_Users/Users.ijm* and thus become visible to others on the network. Note that Sign-in/Startup only affects macro files in the local plugin folder */ImageJ/Macros/_Macros/*, and does not otherwise alter the ImageJ program or file structure. Moreover, after startup, all macros are run locally from */ImageJ/Macros/_Macros/* – files on the network are never called during image analysis.

Multiple accounts and cloud-connected computers can receive updates through network access. While this example shows two connected computers, any number of additional computers can be granted access, as is deemed practical.

Data files generated during image analysis are stored locally on that computer's hard drive on a desktop folder named *OUTPUT ImageJ*. New subfolders added to *OUTPUT ImageJ* can in turn be copied to/or otherwise synced with cloud storage (dotted lines) through cloud client or macro settings.

B.Data Workflow

When a user runs ImageJ with Custom Macros installed, two new buttons become immediately available in the interface. Clicking the question mark button opens either plugin documentation or a list of keyboard shortcuts, depending on menu choice. The 'Start' button to its left (inside the dashed rectangle) initializes the plugin and brings up the User sign-in dialog. The current versions of the plugin components are listed at the top of the dialog. If a network folder is not found, a notice is shown, indicating that Custom Macros is running in local mode. Users can choose a previously made login or create a new one. Results data is saved in user subfolders in the desktop folder *OUTPUT ImageJ*. Logins determine subfolder structure based on whitespace separation, and a standard folder hierarchy recommendation is indicated in the dialog. Users can delete old logins from completed experiments (folders corresponding to those logins will not be affected). A text file named based on the user login and ending in 'Results.txt' is created in the login subfolder. This file, named Melissa Auditory cFos POA Results.txt, in this example, is a simple spreadsheet where analyses results are saved. Upon completion of startup and confirmation of the output folder, users will have access to the image analysis macros. A user will then load an image, in this case ID_A 20x POA 03.tif, and either run a quantitative macro as a next step, or perform some preliminary tasks, indicated by the five pointed asterisk ⋆. In this example, the user first created a rectangular selection over the ventral part of the image, before running the 'Assisted Count with Find Maxima' macro. This procedure creates folder */POA/ID_A 20x POA 03/*, saves a set of files in that folder, and appends a new entry into file */POA/*Melissa Auditory cFos POA Results.txt. If said folder already exists, the user is prompted to either overwrite or create an additional folder with a timestamped suffix. Every quantitative macro likewise creates a new subfolder, saves files in that folder, and appends measurement data to the login's Results.txt. In this case, four images are saved. ID_A 20x POA 03 original.tif is a copy of the original image before analysis. ID_A 20x POA 03 region.tif shows the rectangular selection made by the user. Asterisk ∗ indicates that the next two images were cropped around this selection. ID_A 20x POA 03 count 21.tif is the final analyzed image with the count result of 21 points watermarked in the image and saved in the filename. ID_A 20x POA 03 auto.tif, is red on black rather than multicolor because red is the color channel containing the label of interest chosen by the user. This image contains the final data points, as well, but with smaller labels, and only the channel of interest is included as a counterpoint to the color image for further clarity in reviewing analysis results. The two zip files are region of interest (roi) files: ID_A 20x POA 03 region roi.zip is of the aforementioned rectangular selection and ID_A 20x POA 03 count 21 roi.zip contains point selections for the 21 data points counted. The two csv files are comma-separated values files (simple spreadsheets). ID_A 20x POA 03 region results.csv is measurement data for the rectangular selection and ID_A 20x POA 03 count 21 results.csv is measurement data, including x,y coordinate values, for each of the 21 data points. Other quantitative macros may not save the same number of files, but all will save at least two images, a region of interest zip file, and a measurement spreadsheet. Data appended to Results.txt are formatted with colons as field separators and single quotes as string delimiters.

example, an image called Brain 40x.tif would return a calibration multiplier of 40. The applicable relevance of the calibration multiplier is that it conveniently corresponds to the optical magnification used in an image, providing users with a straightforward method for scaling images taken at different magnifications on the same microscope. The μm^2 to px calibration value is the ratio of a camera sensor's cell size (pixel size) divided by the optical magnification (Sellaro et al., 2013). A microscope camera's cell size can be found in the camera manual or in imaging software settings. If the cell size value is accurate, the calibration multiplier is identical to the magnification. In Custom Macros, cell size value is stored locally in the file */ImageJ/Macros/CustomMacros_Macros/Calibration.ijm* and, upon installation, is adjusted to conform to the equipment used. For example, the Calibration.ijm file for a camera with a 7.4 μm^2 sensor should read 7.45 um. This file only needs to be set once, unless different settings are required. Thus, image file Brain 40x.tif is calibrated as follows: 7.4/40 = 0.185 μm^2 per px. The default value found in Calibration.ijm is 6.45 µm and the calibration multiplier used if no suitable string is found in the filename is 20. A calibration multiplier of 0x bypasses scaling for that image, resulting in measurement output in raw pixels (a scale of 1px per 1px). Filename-Based scaling can be disabled for all images by leaving the value in Calibration.ijm blank. See Supp.Video 2 for a demonstration of Automated Filename-Based Calibration.

3.2.2. Color channel selection hotkeys

Number keyboard shortcuts ('1'–'7') isolate color channels for viewing. This is done by temporarily converting multicolor images into RGB composite stacks. These hotkeys work whenever an image is open as well as during interactive portions of quantitative macro procedures, facilitating user visualization and temporary color leveling for single-color features (e.g., fluorescently labeled cells). Keyboard shortcut '8' returns the image to non-composite RGB, though this is done automatically when a quantitative macro is run.

3.2.3. Grid subsampling

Grid-based random subsampling is a standard method in stereology and general image analysis. Keyboard shortcut 'g' sets up grid dimensions if none are saved. Once dimensions are set up, a shortcut 'g', randomly places a grid of that size over an image. Keyboard shortcut 'shift + G' allows users to review the currently saved grid dimensions and gives the option of resetting them. Grid dimensions are stored in the user login subfolder in file Grid.ijm. New dimension settings need to be created for each new login subfolder, since different image calibration multipliers, anatomical regions, and experiments typically require specific grid sizes.

3.2.4. Context-based filename suffixes

Multiple analyses are often done on the same image or on one with the same name, such as for colocalization measurements. Keyboard

Table 1
Function Overview. A listing of the procedures Custom Macros can run via a single keyboard shortcut or interface button click. Parentheses indicate Results sections in which the respective macro functions are described. Commands under 'CORE INSTALLATION MENU' are available after login. Commands under 'ADDITIONAL MACROS MENU' can be accessed via the 'Additional Macros' interface button (Section 3.5, Supplementary Fig. 1A). Pressing shift, alt, control, or space while invoking some of these commands adds alternative options.

Core Installation Menu					Additional Macros Menu	
Startup	Pre-Analysis Preparation	Image Analysis		Post-Analysis Review	Batch Image Preprocessing	Extra Functions
		Point Count	Area Measurement			
User Login (3.1.4)	Filename-Based Calibration (3.2.1)	Assisted Point Count with Find Maxima (3.3.1.1)	Start of Measurement Analysis (3.3.2.1)	Region and Overlay Review (3.4.1)	Subtract Background Batch (3.5.2.3)	Field of View Overlay (3.5.3)
	Color Channel Hotkeys (3.2.2)	Manual Point Count (3.3.1.2)	Completion of Measurement Analysis (3.3.2.2)	Point reanalysis (3.4.2)	Stack Project Batch (3.5.2.4)	Look-Up Table Batch (3.5.4)
	Grid subsampling (3.2.3)		Assisted Thresholding (3.3.2.3)	Statistical Tests and Histogram Tool (3.4.3)	Enhance contrast batch (3.5.2.5)	Macro Development Mode (3.5.5)
	Context-based filename suffixes (3.2.4)		Threshold Region Creation (3.3.2.4)		Threshold Outline Batch (3.5.2.6)	
	Masking and Cropping (3.2.5)		Area Measurement Batch (3.3.2.5)			
	Random Code Generator (3.2.6)					
	Training Image Generator (3.2.7)					

Table 2
Comparison of Custom Macros Semi-automated Command-based versus Manual Software Interaction Procedures Shown is a step-wise comparison between using Custom Macros' semi-automated single-command shortcut method versus manual interaction with ImageJ menu commands. The procedure is that used for Fig. 3D1 and utilizes the 'Color Channel Hotkeys' macro (Section 3.2.2) and the Start of Measurement Analysis macro (Section 3.3.2.1). The left-hand column lists users' steps with Custom Macros. The right-hand column lists the steps required to achieve the same result manually. Single rows indicate shared actions between the two approaches, namely opening an image and drawing a selection. 'Once per session' indicates that the settings entered into the dialog box invoked by Keyboard shortcut F5 (Fig. 2C) may be saved for the duration of the analysis session, in which case dialog interaction occurs only once until ImageJ is restarted. The point here is that it takes a user 26 manual ImageJ interface commands to achieve the same result Custom Macros automates with two keyboard shortcuts.

Custom Macros commands	Manual ImageJ procedure
File→Open: ID_A 20x OE ChAT.tif	
*(Handled by Keyboard Shortcut **'F5'**)*	Using a folder manager (e.g., Windows Explorer), create directory:...*desktop/OUTPUT ImageJ/User/Experiment/ChAT/OE/ID_A 20x OE ChAT/*
	Analyze→Set Scale, calibrate to 0.3225 µm distance
	File→Save as→Tiff:.../ID_A 20x OE ChAT/ID_A 20x OE ChAT original.tif
Keyboard Shortcut '**3**'	Image→Color→Make Composite
	Image→Color→Channels Tool
	In Channels window, deselect channels 1 and 2
Draw freehand selection around area of interest (blue dendrites)	
Keyboard Shortcut '**F5**'→Dialog Box Entry (once per session)	Analyze→Tools→ROI Manager
	ROI Manager Window Menu→Add
	ROI Manager Window, select added entry
	ROI Manager Window Menu→Rename: ID_A 20x OE ChAT select
	ROI Manager Window Menu→More→Save:.../*ID_A 20x OE ChAT*/ID_A 20x OE ChAT select.zip
	Image→Type→RGB Color
	Edit→Options→Colors, Foreground: yellow
	ROI Manager Window Menu→More→Draw
	File→Save as→Tiff:.../*ID_A 20x OE ChAT*/ID_A 20x OE ChAT select.tif
	Close window ID_A 20x OE ChAT select.tif
	File→Open: ID_A 20x OE ChAT.tif
	Image→Color→Split Channels
	Close window ID_A 20x OE ChAT.tif (red).tif
	Close window ID_A 20x OE ChAT.tif (green).tif
	Select window ID_A 20x OE ChAT.tif (blue).tif
	Image→Rename: ID_A 20x OE ChAT
	ROI Manager Window Menu→Measure
	Results window→Save as:.../*ID_A 20x OE ChAT/*ID_A 20x OE ChAT select.csv
	Results window→Results→Clear Results
	Edit→Selection→Select None

shortcut 'F2' allows users to create a set of saved suffixes for an experiment and easily rename images, as well as delete and create new suffixes. Suffix settings are stored in the user login subfolder in file FolderConfig.ijm and need to be created for each new login subfolder. Batch file renaming is available via keyboard shortcut 'r'. This macro allows users to replace or remove terms in the filenames of all or selected file types in a folder and then copy these to a new folder. Original files are left unchanged.

3.2.5. Masking and cropping

After selecting an area in an image, users may mask the foreground or background in black using custom interface buttons (Supp. Fig. 2A). Clicking the red, left facing mask button fills everything in a selection with black. This is useful for masking artifactual signal prior to quantitative analysis. The blue, right facing mask button fills everything outside of a selection with black. Holding shift while clicking the blue button also crops the masked selection while doing so with the red button inverts the selection outside of the masked region.

3.2.6. Random code generator for experimental groups

Image filenames may sometimes reveal the experimental grouping of subjects, compromising observer blinding during analysis. To create a random code list to correspond to a set of subjects, users press keyboard shortcut 'shift + R' and enter a list of names. The macro will generate a page of random letter codes corresponding to each name (e.g., X = Animal01, P = Animal02, etc.). Entering more than 26 names will generate codes with added letters (e.g., Y = Animal30, GG = Animal02, etc.).

We also recommend using this function for the purpose of generating random sub-sample lists for analysis, as in the following scenario: In an experiment composed of dozens of images, 50 neurons were counted in a given micrograph. It is neither practical nor statistically advisable to manually trace and measure the cross sectional area of all 50 cells (and consequently every other cell in every micrograph in the cohort). Custom Macros assigns sequential numbers to objects during counts. The user would enter that sequence, randomize it, and select the appropriate sub-sample number of cells to measure based on alphabetical order.

3.2.7. Training example image generator

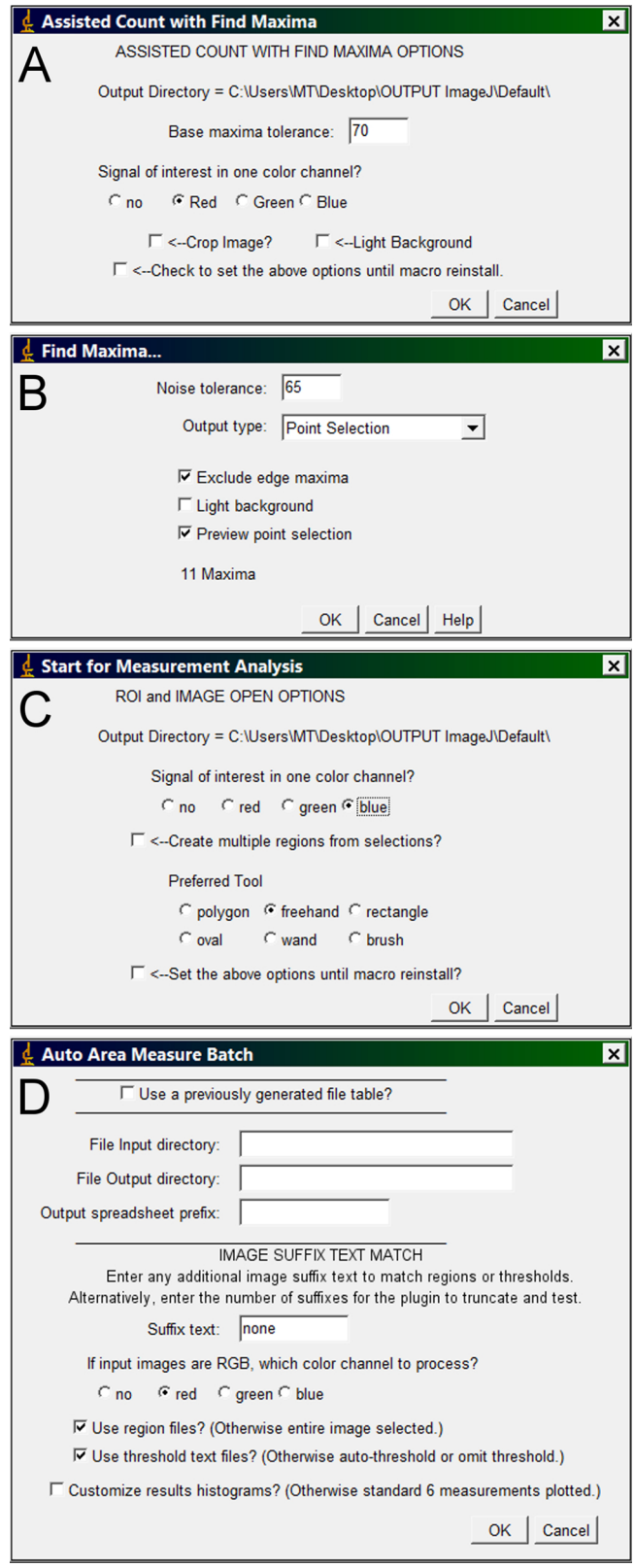
Prior to using Custom Macros on experimental micrographs, some users may find it instructive to first try on simplified images. The Training Example Image Generator macro creates images with contrast features designed to demonstrate the underlying principles and processes that are employed by the quantitative image analysis macros. Keyboard shortcut 'shift + V' calls the macro, which prompts users to set a variety of image parameters relevant to different analysis situations, including RGB/grayscale settings, z stacks, and size dimensions, and then generates that image. Keyboard shortcut 'v' will generate duplicate images with those parameters or a generic image if none were set. Resulting example images are grids with intensity values ideal for demonstrating different thresholding techniques, colocalization, assisted point count fundamentals, and the effects of filters. See Supp. Video 2 for a demonstration of Example Image Generation.

3.3. Image analysis

Of the types of neurohistological image analysis goals that an experiment may involve, most can generally be grouped into two categories based on the nature of the data generated. 1) Point and object counts produce ordinal values of summed discrete data points. The most common example of such analyses is cell counting. 2) Analyses of raster (pixel) space in two-or-higher dimensions generate continuous interval data such as area, volume, and diameter, along with proportions derived from these, like aspect ratio and circularity. For convenience, we will refer to all such measurements as 'area measurement' analyses. Both point and area data in ImageJ also contain the pixel intensities of the region(s) of interest of the underlying image, as well as x,y,z selection coordinates relative to image size. As such, point counts, when comprehensively recorded, can yield more detailed results than simply sums of counts. Therefore, it is worth noting that while the point count and area measurement macros in Custom Macros are performed as separate procedures, resulting data from either task can ultimately inform approaches to and be utilized for diverse statistical ends.

3.3.1. Point count macros

3.3.1.1. Assisted point count with find maxima. The process of count analyses in images with numerous objects can be expedited by using contrast-based segmentation to assist in identifying data points. The Assisted Point Count macro (same as 'Count Macro' in Fig. 1B) interactively utilizes the 'find maxima' process to facilitate this task (Figs. 2A, 3A1–3; keyboard shortcut 'F4'). The find maxima function can be very effective for images with discrete objects and minimal background but is not sensitive to object size or global intensity. Images with issues related to the above limitations may be made more amenable to maxima analysis through the masking of troublesome spots (Section 3.2.5) or pre-analysis background filtering (Section 3.5.2.3). The Batch Particle Analysis macro (Section 3.5.2.6) is an alternative procedure for the semi-automation of point and object counts. See Supp. Video 3 for a demonstration of the Assisted Point Count with Find Maxima procedure.



(caption on next page)

Fig. 2. Image Analysis Dialogs.

A.Assisted Count with Find Maxima Start Dialog

Keyboard shortcut 'F4' invokes the Assisted Count with Find Maxima macro and prompts the dialog shown. The user login folder ('Output Director') is shown at the top and the macro creates a new subfolder named for the loaded image. The find maxima algorithm identifies pixels that are local maxima – relative peaks of intensity compared to their neighbors. Tolerance sets how far apart intensity values are required to be to count as local maxima. In practice, the lower the tolerance, the more maxima are identified and points added. Users set a 'base tolerance' as an automated first-pass attempt for each macro run and dynamically adjust tolerance, as needed (Fig. 2B). Though there are exceptions, the find maxima process is typically run on either single color channels of RGB images or grayscale images. Thus, unless one is working with an 8-bit grayscale micrograph or a suitably uniform color image, the color channel of the RGB image containing the signal of interest ('signal to be counted') needs to be chosen in the dialog. The final checkbox saves all the above settings until ImageJ is restarted or another user logins in.

B.Find Maxima Adjustment Dialog

Users may adjust noise tolerance parameters after evaluating point counts generated by the base tolerance set in Fig. 2A. Note that the output type is automatically set to 'point selection' and must remain so for the macro to work. Checking 'Exclude edge maxima' is recommended.

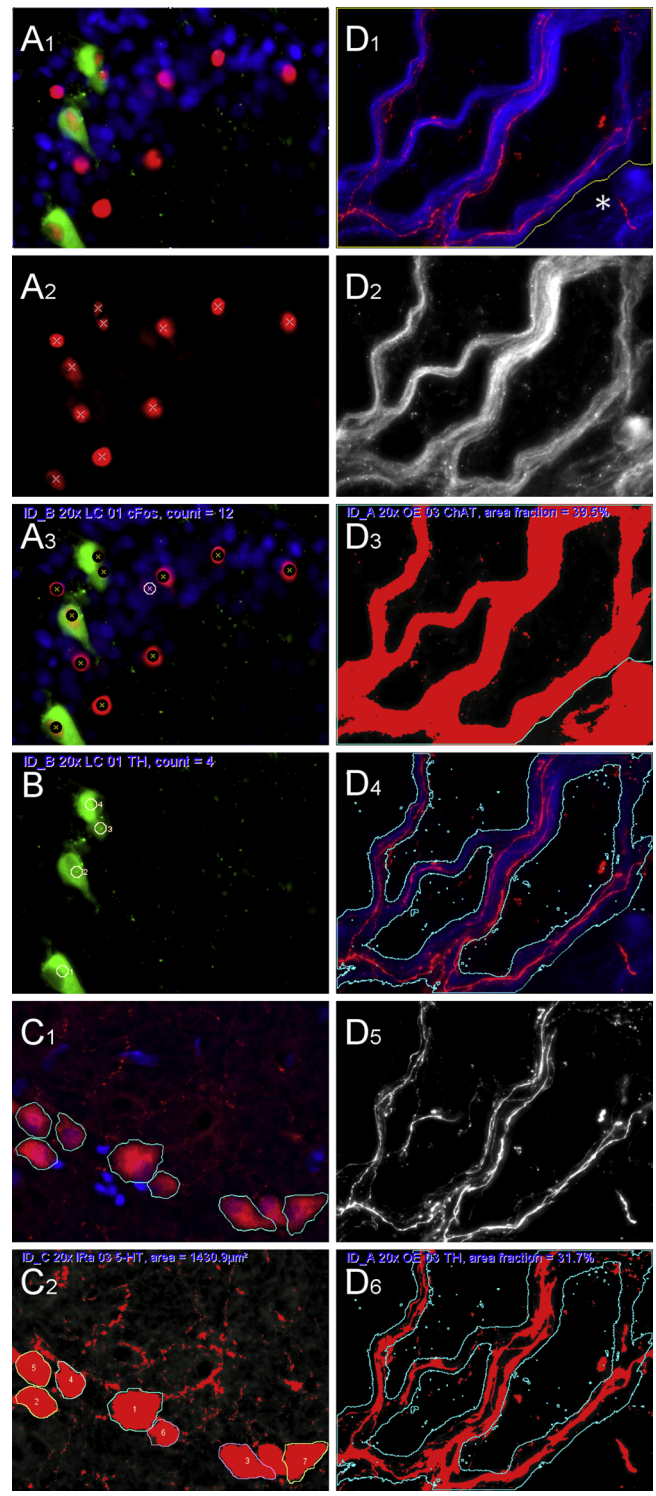
C.Start of Measurement Analysis Dialog

Area measurement is a multi-step process and thus requires running a sequence of macros. The first of these (invoked via keyboard shortcut 'F5') prompts the dialog shown. The user login folder ('Output Directory') is shown at the top and the macro creates a new subfolder named for the loaded image, establishes and saves selection regions, and confirms analysis parameters. As with the dialog box in Fig. 2A, users choose a signal of interest color channel. See Fig. 3C1, D1 for examples of when to select the 'Create multiple regions from selections?' checkbox. A preferred selection tool may also be chosen and settings saved.

D.Area Measurement Batch Dialog

The Area Measurement Batch macro enables the automated signal and area analysis of multiple images without direct user interaction. Keyboard shortcut 'F10' prompts the dialog shown. This macro processes images according to a file table spreadsheet that it generates based on the options chosen and the contents of the specified input folder. Alternatively, a previously created or user-made file table may be used. Region and threshold text files corresponding to images placed in the input folder enable area measurement based on the parameters of those files, which the macro identifies by matching region ('.roi' or '.zip' extensions) and text files ('.txt' extension) whose filenames contain the text of the image filenames. For example, image ID A 4x Thalamus 05.tif is matched both with region ID A 4x Thalamus 05 *something*.zip and threshold text file *ID A 4x Thalamus 05 something else 128.txt*. Thresholds are identified from text file names: any number from 0 to 255 not found in the image name and padded on both sides with spaces or a space on the left and period on the right is considered a threshold. If image filenames have additional text not found in region or threshold filenames, for example, ID A 4x Thalamus 05 TH.tif, 'TH' must be entered in the dialog for matching to work. If there is no additional text in the filename, the entry is left at the default of 'none', as in the figure, or can be left blank. File matching discrepancies, such as missing or mismatched filenames, are logged in the file table and reported to the user and corresponding images are omitted from batch analysis. Regions and threshold text files saved during semi-automated area measurement procedures follow this naming scheme. Thus a key use for this macro would be the batch analysis of a cohort of experimental images using regions and thresholds corresponding to different markers (from different color channels). Running the procedure generates and saves three spreadsheets: 1) The file table of images, regions, and threshold text files found in the input folder. 2) The batch analysis measurement results. 3) An abridged list of core area measurement results in the format of results.txt files generated by all image analysis macros (see Fig. 1B legend). Spreadsheet files are named based on the 'Output spreadsheet prefix' option, with user login name being the default. Two versions of each image analyzed are saved: 1) A copy of the original. 2) A post-analysis image with region and thresholds outlined (as in Fig. 3D4, D6). Finally, frequency distribution histograms are generated and saved as a montage image. Users may specify which results metrics are used to calculate histograms.

3.3.1.2. Manual point count. The complex cytomorphology of neural tissue often makes automation of object detection unreliable-to-impossible in our experience, regardless of the sophistication of the segmentation algorithm methods attempted. Instead, point counts may be performed manually through visual identification of objects by users (Fig. 3B; keyboard shortcut 'F3'). Both kinds of point count macros allow users to toggle between color channels and temporarily adjust intensity and gamma levels for the purpose of highlighting morphological features and lowering background and artifactual signal (Section 3.2.2).



(caption on next page)

Fig. 3. Image Analysis Example Micrographs.

(A1–A3)Assisted Count with Find Maxima.

A1. After an image is loaded, a selection may be created surrounding the part of an image to be counted. Once the macro is called with keyboard shortcut 'F4', this selection is saved as a region for later review. The area outside the region will be masked out in black and, depending on the checkbox setting in Fig. 2A, also cropped. Shown is an RGB immunofluorescence micrograph at the start of analysis. Red is label for cFos immunoreactivity (-ir), a nuclear immediate early gene product associated with neuronal firing. This is our signal of interest to be counted, which is why red was selected in Fig. 2A. Green is tyrosine hydroxylase-ir, a cytoplasmic protein that is the rate-limiting enzyme for catecholamine synthesis. Blue is DAPI nuclear counterstain. Since the user did not create a selection in this image, its entirety is selected as the analysis region of interest (the region outline is visible as a white line surrounding the image).

A2. After optional adjustment for noise tolerance level (Fig. 2B), find maxima identified 11 points, marked as white X's. At this point, the user can remove any inaccurate points chosen by the algorithm.

A3. Points chosen by find maxima that were not removed by the user are masked with black ovals inscribed with a yellow X. Users now add additional points (here a white circle inscribed with a white X, indicating a single addition). After user confirmation, count data is marked in the upper-left corner, output data saved, and the macro ends. Note that Fig. 3A1–A3 are the actual images saved in the user subfolder (as in Fig. 1B).

(B)Manual Point Count

B. Due to the comparatively diffuse nature of green signal and the small number of labeled cells, the Manual Point Count macro is chosen to count objects in the green channel of the image from Fig. 3A1–3. The user, having chosen the four cells based on morphological criteria, clicks on each, and presses keyboard shortcut 'F3' to comple the procedure. Points counted are indicated by numbered white circles. Numbering is optional and can be configured via keyboard shortcut 'F2'.

(C1–C2)Area Measurement of Multiple Regions.

C1. It is often necessary, or at least expedient, to measure more than one region of a micrograph in a single pass, as for example when analyzing the cross-sectional areas of multiple cells in an image. In this case, the user must select the 'Create multiple regions' checkbox in Fig. 2A. Shown is an RGB immunofluorescence micrograph in which red is label for serotonin-ir, a neuromodulator expressed in cytoplasm, and blue is DAPI nuclear counterstain. Serotonin being the measurement signal of interest, the red channel was selected in the dialog in Fig. 2A. The user traces several neurons based on cytomorphology, adding each selection as a region, and pressing 'F5' when complete. The pixel intensity underlying regions can be measured with or without thresholding, depending on the goal of the analysis. If thresholding is not needed, the user presses keyboard shortcut 'F8' to complete area measurement analysis.

C2. In our example, thresholding is utilized so the user presses keyboard shortcut 'F6' calling up the Assisted Thresholding macro. A red threshold overlay, calculated by auto-thresholding algorithms, appears over the image. Thresholds are, by default, inclusive. Threshold minimum (histogram black point) can be toggled up or down using keyboard shortcuts 'x' and 'z', respectively, or via the threshold dialog. Keyboard shortcut 'c' toggles threshold visibility so that the user can review the underlying non-thresholded image. Upon establishing a satisfactory threshold minimum value, the user presses keyboard shortcut 'F8' to complete area measurement analysis. Thresholded area is measured for each region, and output data is saved, including the image shown. The μm^2 value in the watermark is the summed cross-sectional area for all seven regions.

(D1–D6)Area Measurement with Threshold Region Creation for Colocalization

The goal of the analysis in Fig. 3D1–6 is to determine the amount of overlap (colocalization) between two color channels in a selected anatomical location. Since threshold region creation and colocalization only work for single region selections, the 'Create multiple regions' checkbox must be unselected in the dialog in Fig. 2A.

D1. Shown is an RGB immunofluorescence micrograph in which red is label for tyrosine hydroxylase-ir and blue is choline acetyltransferase-ir, an enzyme involved in the synthesis of acetylcholine and thus a marker for cholinergic cells and fibers. Of anatomical relevance in the context of this analysis are the thick cholinergic fiber tracts diagonally traversing the image frame from the lower-left to upper-right. However, the frame also contains cholinergic somata and fibers from a neighboring but functionally unrelated anatomical nucleus (asterisk in the lower-right) that needs to be excluded from measurement. The user draws a selection around the relevant anatomy, excluding the area in the lower right, and presses keyboard shortcut 'F5'. The region created from that selection is indicated by the yellow outline. Note that for the purpose of presentation clarity, the outline was offset by 1 pixel at the edges.

D2. Shown is the blue channel (in grayscale) isolated by the previous macro. The Auto Threshold and Measurement macro (keyboard shortcut 'F6'), previously described for Fig. 3C2 is run to threshold blue channel signal.

D3. With an appropriate threshold set, the user runs the Threshold Region Creation macro (keyboard shortcut 'F7'), which saves a region zip file of the thresholded area as bound within the previously delineated selection (yellow outline in Fig. 3D1). Keyboard shortcut 'F8' completes area measurement analysis. Shown is the resulting saved image with area above threshold in red and the region selection outlined in cyan. The marked text contains the 'area fraction', the percentage of the region selection that is above threshold.

D4. Continuing colocalization analysis, the user loads the same image. Importantly, the region threshold zip file from the previous analysis must also be loaded. Using the Suffix Namer macro (keyboard shortcut 'F2'), the image is renamed with a suffix to reflect the signal to be analyzed (in this case TH for tyrosine hydroxylase). From here, the area measurement analysis begins as before (initiated via keyboard shortcut 'F5'), except with red selected as signal of interest instead of blue. Shown is the resulting saved image with the loaded region threshold selection, indicating relevant cholinergic fibers, outlined in cyan.

D5. Shown is the red channel (in grayscale) isolated by the previous macro. The Assisted Thresholding (keyboard shortcut 'F6') is again run.

D6. The user repeats the procedure described for Fig. 3D3. The final saved image shown informs the result of the complete analysis – the red threshold overlay within the cyan outline indicates the location of colocalized red and blue signal. Here, 'area fraction' in the marked text indicates the percentage of tyrosine hydroxylase label covering the anatomically relevant cholinergic fiber tracts. As with all quantitative macros, saved in the image subfolder is a csv spreadsheet of measurements (e.g., mean, median, standard deviation, skew, etc.), in this case within the colocalized area.

3.3.2. Area measurement macros

Unlike point counts, area measurement analyses require a sequence of macros to be run. This is due to the importance of establishing the desired regions of interest within an image and because of considerations of proper signal thresholding. Thresholding is not required for area measurement but is usually recommended. Measurement of multiple selections in a single image is supported, as long as colocalization is not required.

Colocalization is essentially the overlap of two color channels in a selected region of an image. This overlap informs potential functional relationships between molecular targets being investigated, depending on the hypothesis of the study. Though colocalization research often focuses on double-labeling of the same cell, in our context the term also applies to intercellular overlap of signal. For example, the presence of beaded fiber neurotransmitter label on histochemically identified motor neuron somata may suggest synaptic interaction (Rosner et al., 2018). Colocalization analysis is achieved by running area measurement macro sequences for separate color channels of the same image. See Figs. 2C, 3C1-2, D1-6, and Supp. Video 4 for a detailed description of area measurement procedures.

3.3.2.1. Start of measurement analysis. Every area measurement analysis requires the Start of Measurement Analysis macro (keyboard shortcut 'F5') to first be run on a loaded image (Figs. 2C, 3 C1, D1, D4). This command sets analysis parameters and saves the analysis region and its measurements.

3.3.2.2. Completion of area measurement analysis. Every area measurement

analysis likewise requires the Completion of Area Measurement Analysis macro (keyboard shortcut 'F8') to be run as a last step to complete the procedure (Fig. 3C2, D3, D6). This command saves a final measurement spreadsheet, appends measurement data to an experiment's results tables, and creates and saves an image and region file representative of analysis results. If thresholding is not required, this macro is called after the step from the previous section. Thresholding is usually required for most experiments, so the following two macros should be run prior to the completion macro.

3.3.2.3. Assisted thresholding. This macro (keyboard shortcut 'F6') runs a combination of auto-threshold algorithms that we found work well for neural tissue. Users may either accept the automated threshold or toggle threshold minimum based on visual appraisal. Threshold can be set using keyboard shortcuts 'x' and 'z' to raise or lower the minimum value stepwise, or via slider. The solid red overlays in Fig. 3C2, D3, D6 are thresholds.

3.3.2.4. Threshold region creation. This command (keyboard shortcut 'F7') saves a region file of the threshold selection and a text file containing measurement data with the threshold value in the filename. Generating regions from thresholds is useful in a variety of contexts, and running the Threshold Region Creation macro is recommended for any area measurement analysis that does not involve multiple regions (see Fig. 3C1-2 for this exception). The text files, aside from providing a record of the data, may be used to batch process images with saved thresholds (Section 3.3.2.5). Note that Threshold Region Creation is necessary for colocalization analysis (Fig. 3D1-6).

3.3.2.5. Area measurement batch. Area measurement can be automated as a batch function for a folder containing multiple images. Keyboard shortcut 'F10' calls up the Area Measurement Batch macro dialog for users to enter the relevant parameters (Fig. 2D). Region and threshold files generated in the preceding procedure (Sections 3.3.2.1–3.3.2.4) may be used in conjunction with this macro, though they are not required. Area Measurement Batch macro uses include: (1) Automation of colocalization analysis. (2) Batch measurement using auto-thresholds, threshold values loaded from a spreadsheet table, or without thresholding. (3) Combined use with Batch Image Preprocessing macros (Section 3.5.2). (4) Testing, replication, or augmentation of previous area measurement analyses.

3.4. Post-analysis review

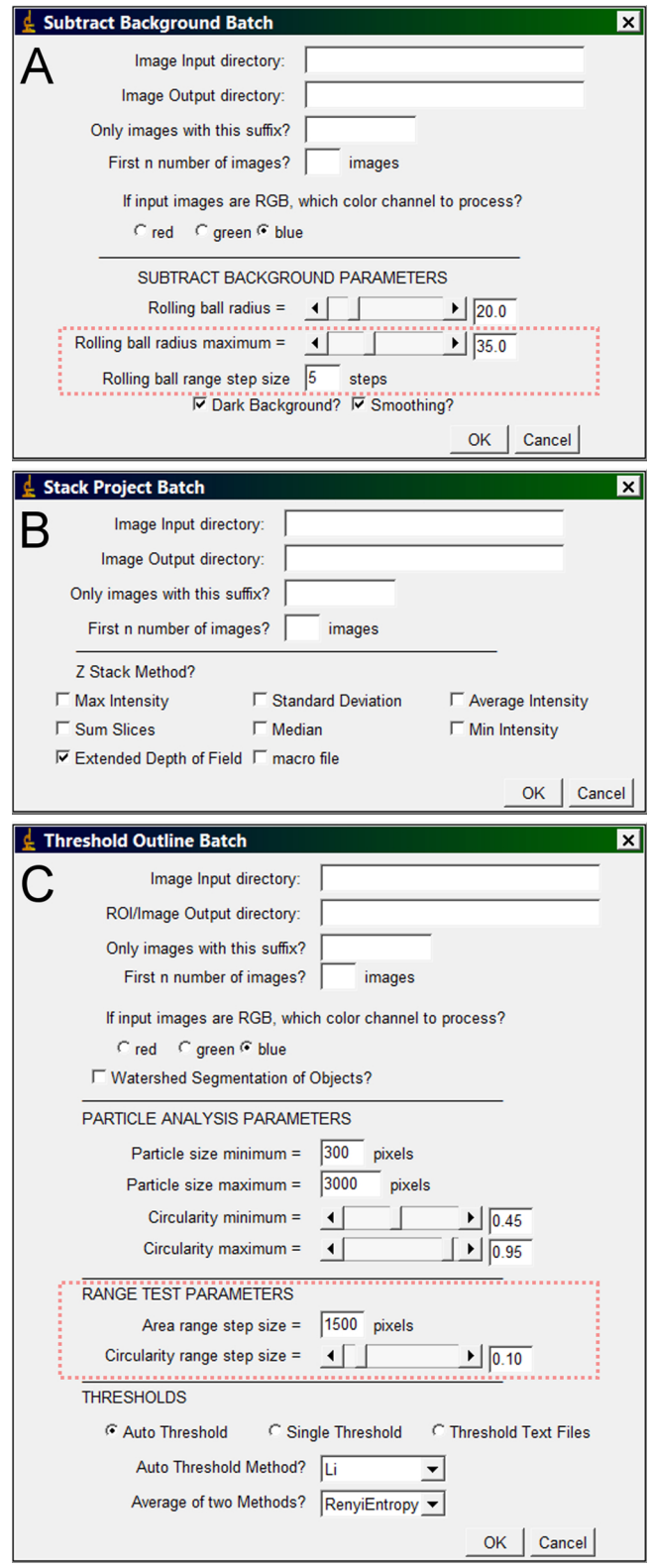
3.4.1. Region and overlay review

Regions created by Custom Macros may be loaded after quantitative analysis, typically to review over the original unmarked version of the analyzed image. Since both a region file and the original image are saved in a subfolder after every quantitative macro run, such a review is straightforward to perform. With the original image loaded, keyboard shortcut '9' will open the region file (zip or roi extension) found in the parent folder of that image. If multiple regions are found, the last modified region file matching the base image filename is loaded. An error is returned if no region is found. Visibility of regions is toggled on and off using keyboard shortcut '0'. Loaded regions can in turn be converted to overlays with keyboard shortcut 'o' and their visibility toggled on-and-off using keyboard shortcut 'i'.

3.4.2. Point reanalysis

One application of the Region and Overlay Review macro in the above section is the correction of point counts in previously analyzed micrographs. After being loaded, a region file containing point count data can be converted back to point selections with keyboard shortcut 'p'. Additional points can then be added, moved, or removed, and the Manual Count Macro run to measure and record updated point results (see Section 3.3.1.2). As with all Custom Macros quantitative procedures, users are prompted if re-running a count would overwrite existing results files, and are given the option to either proceed and overwrite or create a new folder. This check prevents the accidental deletion or modification of data.

We suggest a best practice approach of creating a new user login when reanalyzing points so as to direct output to a separate folder from



(caption on next page)

Fig. 4. Batch Preprocessing Dialogs.
All Batch Preprocessing dialogs have the following features: text entry boxes for input and output directories; a text entry box for limiting processing to only a subset of images ending with a specified suffix; and a numerical entry box specifying that only a subset of images within a directory should be processed (typically used for testing). The Subtract Background and Threshold Outline Batch macros also have 'range test modes' that allow users to test a specified range of settings, generating files for each step of that range. Range testing enables users to identify the most appropriate batch settings and is available as one of two options available upon pressing the respective command button.
A. Subtract Background Batch Macro Dialog
Subtract background works on 8-bit single plane images and stacks and allows for specifying the color channel for processing RGB images. Rolling ball radius is typically set to the largest object in an image. The dialog shown is running in range test mode with range functions outlined in red. At the range and step shown, four processed images will be generated per input image at radius settings 20, 25, 30, and 35.
B. Stack Project Batch Macro Dialog
The user chooses the z-projection method with which to stack input images. If Extended Depth of Field (EDF) is selected, an additional dialog will request entry of EDF quality setting.
C. Threshold Outline Batch Macro Dialog
The ImageJ particle analysis function is a segmentation method based on size, shape, and threshold. The purpose of this macro is to save region of interest (roi) zip files of the outlines of objects identified based on the chosen particle analysis parameters. Images with drawn outlines are also saved. Watershed segmentation attempts to separate discrete objects in connected clumps. Users choose minimum and maximum sizes and circularity values as criteria for defining objects. Thresholding is required for establishing object boundaries for segmentation and users have three choices to establish thresholds: (1) Auto-Threshold, (2) Single Threshold, and (3) Threshold Text Files. (1) The Auto Threshold choices are listed from more-to-less inclusive, based on our tests. A second threshold method can be chosen to produce an average of the two methods. (2) The Single Threshold option prompts the user to enter a minimum value that will be applied to every image. (3) The Threshold Text Files option applies the threshold values found in the filenames of text files in the input folder (see legend for Fig. 2D for an explanation of the threshold text file format). The Threshold Outline Batch macro works on RGB and 8-bit single plane or stack images and allows for specifying the color channel for processing if RGB images are used. The dialog shown is running in range test mode with range functions outlined in red. At these settings, 10 processed images and regions will be generated per input image: five from 300 to 1800 pixels with circularities of 0.45 – 0.95, 0.55 – 0.95, 0.65 – 0.95, 0.75-0.95, and 0.85 – 0.95, and another five from 1800 to 3000 pixels at the above circularities.

the original. For example, if the initial analysis was done under login 'Melissa Auditory cFos POA', the reanalysis login 'Melissa Auditory cFos POA_final_review' will maintain the experiment-specific folder hierarchy (in directory *Output ImageJ/Melissa/Auditory/cFos/*), thus allowing for convenient comparison with the prior attempt while also making clear the status of the new folder content data via suffix '_final_review'.

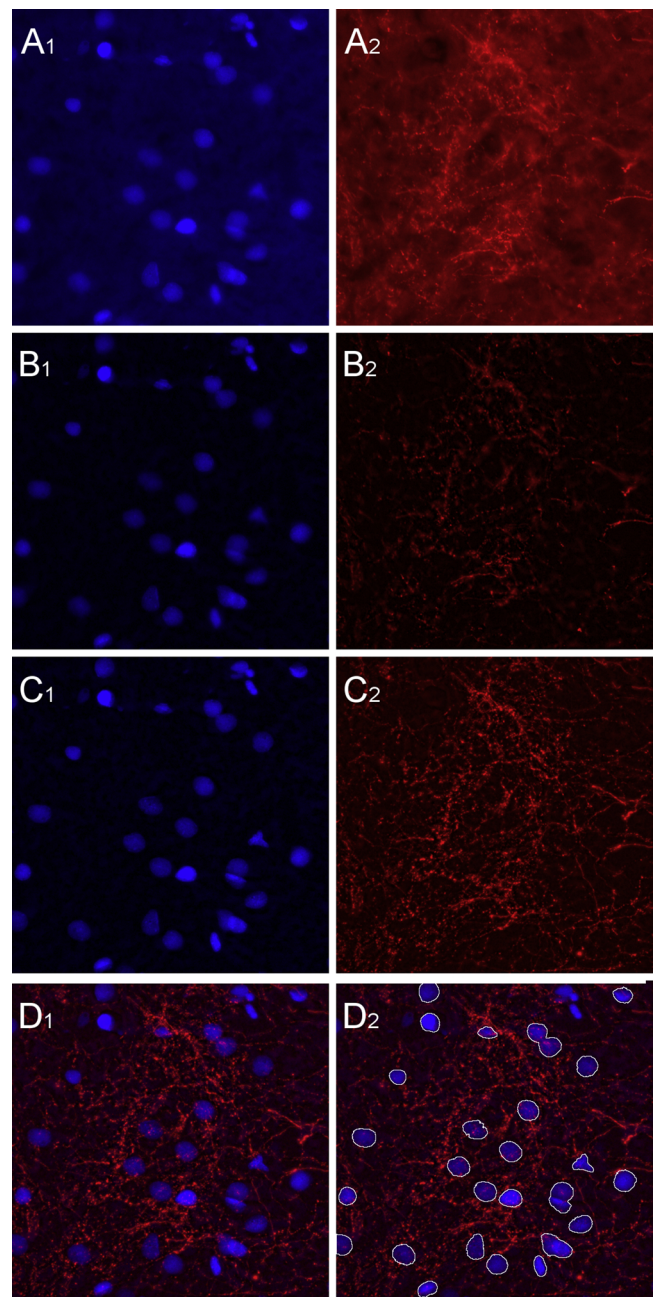
3.4.3. Statistical tests and histogram tool

The histogram tool, invoked via keyboard shortcut 'h' generates a montage image of a set of color histograms derived from open results windows or data tables. Users choose which data columns to quantify and the directory to save the histogram image. The command automatically excludes data columns that cannot be binned and proportionally sizes the output montage image based on the number of histograms needed. The Rice Rule is the default bin formula, but users may set their own number of bins or choose another method, including Scott's, Sturges', and square root formulas. A suite of t-distribution-based statistical tests is also provided and is invoked via keyboard shortcut 'shift + H' or through the Additional Macros menu (Section 3.5). Statistical testing selectes and saves data in the same manner as the histogram tool.

3.5. Additional Macros

3.5.1. Background and installation

Not all macros are installed upon startup and login. This is done to prevent cluttering the core interface with seldom-used functions and to avoid potential performance issues from loading too many scripts and variables into memory. These 'Additional Macros' are stored in the local folder */ImageJ/Macros/CustomMacros_Macros/AdditionalMacros/* and are updated along with the core analysis macros. Additional Macros may be added to the network by the administrator by copying files to the */AdditionalMacros/* cloud folder or locally by users to the aforementioned folder of their ImageJ installations. The Additional Macros approach provides a convenient method for easily and unobtrusively expanding the functionality of Custom Macros. Pressing the 'orange box' interface button (Supp. Fig. 2A) pulls down a menu showing the current Additional Macros installed and allowing the user to select which to run. New interface buttons are added, depending upon which



(caption on next page)

Fig. 5. Batch Preprocessing Example Micrographs.
(A–D) Batch Preprocessing Sequence Example.
Shown are two 8-bit color channel images from a stack micrograph taken of tissue containing DAPI nuclear counterstain (blue) and immunolabeled with serotonin (red). These images were run sequentially with three Batch Preprocessing macros, meaning that Fig. 5A1 was used as the input for Fig. 5B1, etc.
A1–2. Shown are unprocessed, single slice images from the original raw stack micrograph.
B1–2. Subtract Background Batch Macro
Shown are the same slices from Fig. 5A1–2 after the original stacks were preprocessed with Subtract Background Batch macro. The rolling ball radius was set at 30 for B1 and 7 for B2.
C. Stack Project Batch Macro
The subtract background processed stacks from Fig. 5B1–2 were then z-projected using the Stack Project Batch macro. C1 was processed with Extended Depth of Field (quality setting 4) and C2 was stacked using Max Intensity.
D. Threshold Outline Batch Macro
The z-projected images from Fig. 5C1–2 were then merged into an RGB image using the ImageJ 'Merge Channels' command. The Threshold Outline Batch macro was then run on the blue channel with a particle range size of 300–1000 and a circularity range of 0.5–1.0 using Li auto-threshold. The input image is shown in D1. The resulting image, with DAPI nuclei outlined in white, is shown in D2. Note that only blue signal above the threshold minimum and that met size and shape requirements was outlined. Region files of these outlines were also saved, allowing for their potential use in area measurement macros (see Section 3.3.2).

macro is chosen (see, for example, Supp. Fig. 2B). Core image analysis procedures are not available when Additional Macros are installed but pressing the 'Start Button' allows users to re-enter core image analysis mode (Supp. Fig. 2B). At present, four Additional Macros are available, the most significant of which is the Batch Image Preprocessing suite, which will be dealt with first.

3.5.2. Batch image preprocessing

Batch processing is defined as the performance of some procedure(s) on a large number of files (in this case, images). Preprocessing is used here to mean an image manipulation or other task that is run prior to analysis that either changes the content of that image (e.g., background subtraction), or extracts information from that image that is later used to guide the analysis process (e.g., particle analysis). Installing Batch Image Preprocessing adds four new buttons to the interface that correspond to the four batch functions (Supp. Fig. 2B). Upon first running a preprocessing batch function, the user is prompted to choose whether output images should be saved, and in which format (TIFF or PNG). This setting is saved for the remainder of the session, but can be changed by invoking keyboard shortcut 'f'.

3.5.2.1. Rationale for batch preprocessing. The base ImageJ installation includes numerous image processing functions, including contrast enhancement, deconvolution filtering, shape analysis, segmentation, and z-plane rendering. A vast number of processing plugins created by user-developers and institutions are also available for installation through the ImageJ website. Manipulation of experimental images during analyses raises a number of issues: 1) Running these tools and functions from the ImageJ menu is time consuming for the user, especially when multiple functions are involved. Moreover these functions' interfaces often present numerous and complex options. Repeated manual preprocessing of the hundreds of images that make up an experiment results in a non-trivial resource drain and makes user error likely. Compounding this, some more advanced processing algorithms may take a relatively substantial amount of computing time to complete. 2) Image preprocessing manipulations used in image analysis research are an inherent component of an experiment's methodology and, as such, should be documented. In line with the core theme of our plugin, an emphasis on record keeping facilitates reviewing for consistency in analysis procedures and ultimately ensures methodological reproducibility (Goodman et al., 2016). Batch Preprocessing macros serially log all processing details in the image metadata and generate a text record. Whether filters were used, at what settings, on which images, etc., are important facts to be aware of. 3) Micrograph images are essentially the raw data for neurohistological research (Cromey, 2012). Image processing prior to analysis is often a necessary step in order to extract meaningful information from tissue. Thus, if a treatment is applied to images, it seems scientifically prudent, in most contexts, that all images in a sample be likewise processed in as consistent a way as possible. Ad hoc, case-by-case image manipulation by observer analysts may ultimately introduce inappropriate and potentially confounding variation to results and should be minimized whenever possible. 4) The Threshold Outline Batch function (Section 3.5.2.6) provides a powerful semi-automated method for rapidly counting cells, fiber puncta, and other objects in multiple images while also adhering to the principles above by thoroughly saving result and parameter records for later review.

3.5.2.2. Technical aspects of batch preprocessing macros. The commands and procedures run by the Batch Preprocessing macros are well-documented in the ImageJ user guide online, which is annotated with references to papers defining relevant algorithms and theoretical concepts (Ferreira and Rasband, 2012). Our goal here is to describe how users interact with these macros and also to provide an overview, with examples, of batch preprocessing applications in neurohistological research. We refer readers to the above-mentioned documentation, as well as to the wealth of ImageJ resources online, for more technical detail underlying the utilized functions.

3.5.2.3. Subtract background batch. Excessive or uneven background signal in micrographs may be due to issues with the preparation of the specimen being imaged, such as label having weak signal (leading to signal-to-noise issues), non-specific probe binding, or autofluorescence. Background may also result from problems arising during image acquisition related to exposure or optical artifact. High background may be problematic in different ways, depending on the kind of analysis performed. Lack of signal contrast due to background may make difficult the positive identification of objects during point counts. Similarly, background may drown out true signal during area measurement and make proper thresholding impossible. The subtract background process uses a rolling ball algorithm that takes into account the size of the objects of interest in a micrograph. A range test mode is available as one of two options available upon pressing the Subtract Background Batch button. Range testing runs the subtract background procedure in steps within a user-specified minimum-to-maximum range of rolling ball parameters, generating processed images for each step, thus enabling users to identify the most appropriate setting. See Fig. 4A for dialog parameters and Fig. 5B1-2 for micrograph examples.

3.5.2.4. Stack project batch. An image acquired as layered 'stack' in the z-plane must be combined into a two-dimensional projection prior to traditional image analysis. Custom Macros does not currently include implementations for three-dimensional analysis, so z-projection is an important consideration in preparing image stacks for analysis. Several projection methods are natively supported in ImageJ. Trying different projection methods on stacks generated by the Training Example Image Generator macro (Section 3.2.7) demonstrates the different results from each approach. Aside from the standard z-projection methods, Stack Project Batch macro supports the Extended Depth of Field plugin, which produces z-projections with superior focus (Forster Brigitte et al., 2004). Note that this plugin must be installed in addition to Custom Macros prior to use (ImageJ1 Plugins, 2018). See Fig. 4B for dialog parameters and Fig. 5C1-2 for micrograph examples.

3.5.2.5. Enhance contrast batch. This macro uses either histogram stretching or histogram equalization to enhance contrast and may be

of benefit in improving the definition of objects. See ImageJ documentation for details on setting parameters.

3.5.2.6. Threshold outline batch. This macro is a batch implementation of the ImageJ particle analysis process with additional features and options for data generation and review. As in the Subtract Background Batch macro (Section 3.5.2.3), a range test mode is available as one of two options available upon pressing the command button. Range testing runs the particle analysis procedure in steps within a user-specified minimum-to-maximum range of morphometric parameters. For each image in a folder, regions and processed output images are generated for every size and circularity range combination. Output files are annotated with suffixes indicating the settings used and the number of particles counted. A text file detailing the results of the procedure is also saved. See Fig. 4C for dialog parameters and Fig. 5D1-2 for micrograph examples.

3.5.3. Field of view overlay

It is often helpful or necessary to take images of the same histological section at different magnifications. Reasons for this include establishing anatomical boundaries at low magnification prior to high magnification sampling, or creating side-by-side figures for manuscripts. The Field of View Overlay macro utilizes the 'Stitching plugin (Preibisch et al., 2009) to derive image registration parameters to generate, label, and save a side-by-side montage from two open images with an overlay grid indicating the location of the higher magnification image. Upon running the macro, a dialog allows users to enter magnifications for the two images and whether image registration should be calculated from a single color channel or their average. Note that the Stitching Plugin must be installed in addition to Custom Macros prior to use ("ImageJ1 Plugins," 2018).

3.5.4. Look-Up table batch

Look-up tables (LUTs) are specialized palettes of colors corresponding to gray values that can be used to highlight distinct features in an image, detect artifactual signal, or otherwise apply pseudo-coloring. The Look-Up Table Batch macro applies all the LUTs found in the ImageJ/luts/ folder to an opened image and displays the result as a stack. Each stack-slice is renamed with its applied LUT as a suffix. An option also exists to first invert each LUT. The user is thus able to review the result of a large number of LUT applications on a given image and choose one most appropriate for a given task.

3.5.5. Macro development mode

A Macro Development Mode is accessible either through the Additional Macros command menu or by holding alt, control, or space while clicking the start button. This mode facilitates the user development of ImageJ macros by adding an interface button that installs and saves clipboard contents as macros. Shift-clicking this button runs the contents of the clipboard. Up to ten saved macros are stored before being overwritten.

4. Discussion

4.1. Automation, semi-automation, and bias in image analysis

It is important to note that our use of the term 'automation' is contextual — our software is not intended to completely automate the process of neurohistological image analysis independent of observer interaction, such as through the implementation of predictive algorithms. While some aspects of Custom Macros are indeed automatically handled by our software (e.g., image and data saving and calibration), the core image analysis functionality may best be considered 'semi-automated', meaning that varying levels of user interaction are required. For example, the Assisted Point Count with Find Maxima macro (Section 3.3.1.1) may, depending on the tissue and experiment, in practice be considered highly automated if maxima approximation is sufficiently and consistently accurate. Since this is rarely the case with real-world tissue, it follows that an observer must at least monitor the points identified and be able to correct erroneous selections as conveniently and expeditiously as possible. Even if a completely automated approach is adopted and potentially subjective decisions are handled algorithmically, as can be done with the Automated Area Measurement Batch macro (Section 3.5.3), it would be imprudent to accept the resulting data without a human observer afterwards assessing each image analyzed for artifactual thresholding and other issues.

Thus, a level of human subjectivity is inevitable in all but the simplest analysis tasks. While the goal of a fully automated and putatively objective, computerized, artificial intelligence (AI)-driven method to image analysis is attractive in the abstract, the subtly variable nature of histological staining and neuronal morphology require nuanced decision-making that precludes the feasibility of a generalized algorithmic method without initial a priori fine-tuning for each experiment. Research comparing several automated cell count methods revealed "lower detection rates and higher false-positive rates than are acceptable for obtaining valid estimates of cell numbers" supporting the need for manual decision-making regarding object identification (Schmitz et al., 2014). Moreover, while carefully designed human-assisted analyses may still result in a degree of imprecision, such error is typically detectable statistically. Incautious a priori automation could lead to analytical bias and skewed results that nonetheless appear statistically valid (Geuna, 2000). In any case, decisions requiring anatomical considerations, such as the demarcation and region tracing of brain nuclei, simply must be handled manually (see, for example Fig. 3D1-6).

Design-based stereological techniques have been developed to address bias in object count and volume estimation analyses (Schmitz and Hof, 2005; Hosseini-Sharifabad and Nyengaard, 2007; Zhu et al., 2015). Although the object count examples we provide (Fig. 3A–C) would be considered traditional method-based stereology (as in Ekström et al. (1992)), we are keenly interested in incorporating macros facilitating advanced design-based techniques into future versions of our software, especially for the measurement of cell volume. Indeed, Custom Macros currently includes a grid command (Section 3.2.3) that functions as a design-based stereological optical fractionator (Horn and Rasia-Filho, 2018). However, design-based count methods are not appropriate for all experiments. Notably, species with smaller brains tend to contain anatomical nuclei with too few cells to make grid fraction sampling for counts practical. For example, the locus coeruleus group of the species shown in Fig. 3A, B contains less than 200 neurons total (bilaterally) per animal (Ghahramani et al., 2015).

4.2. Neurohistological use cases

We present several examples of quantitative neurohistological use cases of Custom Macros in Fig. 3. These include semi-automated and manual object counts, cell and fiber measurement with thresholding, and colocalization of neurotransmitter label in defined anatomical regions. Research utilizing the Assisted Point Count with Find Maxima macro has recently been published (Forlano et al., 2017; Mohr et al., 2018), as has research using both semi-automated and Manual Point Count macros (Ghahramani et al., 2018). Additional manuscripts utilizing both Point Count and Area Measurement macros are currently in preparation.

The examples shown are certainly not exhaustive in demonstrating the potential uses of Custom Macros and we anticipate novel and diverse applications of our plugin in the future, particularly in the study of brain asymmetry. Notably, since all the quantitative macros generate and save calibrated coordinate data, there is the interesting potential for plugin use in geometric morphometric techniques (Zelditch et al., 2004).

4.3. Benefits to alternative image analysis approaches

We have encountered methodological issues in publications utilizing neurohistological image analysis that can be summarized into four categories: 1) Counting and measurement procedure reporting is vague or practically absent. While the use of dedicated image analysis software like ImageJ is typically stated, analysis and record keeping are presumably performed through ad hoc interaction with individual software commands and plugins (see Table 2). 2) Analysis is performed during microscopy (through the oculars), rather than on acquired micrographs. 3) Images are acquired, but micrographs are analyzed with photo-editing software like Adobe Photoshop. 4) Proprietary, closed source programs, plugins, and scripts are used.

These four issues are problematic in different ways: 1) Inadequate descriptions of how analyses data were acquired make even the most compelling experimental findings less meaningful and the results of such research difficult to replicate. 2) While camera cost concerns are understandable, imaging setups for microscopes are more affordable than ever. In any case, it is hard to justify an approach to quantitative histological analysis that leaves no visual record for verification, is highly dependent on observer visual acuity and subjectivity, and relies on the manual transcription of data. 3) While some image analysis tasks can certainly be accomplished using general-purpose software like Photoshop or GIMP rather than dedicated scientific applications, using specialized software offers substantial advantages and is required for all but the most basic tasks. 4) There are numerous caveats to using proprietary programs for quantitative image analysis experiments, as alluded to in the introduction. Licensing requirements for multiple computers, often enforced with hardware 'dongles', limit lab productivity. High cost makes the purchase of proprietary programs solely for the purpose of systematically reproducing an experiment a low-priority investment for many investigators (Eliceiri et al., 2012). Operating system compatibility and upgradeability is typically limited and support is often paradoxically lacking compared to open source solutions that have much larger user-bases and online communities. Most importantly, access to code fosters transparency in research, and mitigates the potential for 'black box' software methodology.

How does Custom Macros help address these issues? 1) Our software is designed to be simple to use and its procedures straightforward to describe. A vague methodological statement like "cell area measurements were performed in ImageJ," has little meaning to readers of a research manuscript. A write-up of work done with Custom Macros could instead describe the grid or landmark-based region sampling approaches used, batch filters applied, tracing and thresholding macros run, and most importantly the way those macros were used in generating area data. 2) The benefits of modern digital image analysis are difficult to ignore. The ease of use, step-based analysis approach, automated data output, and the fact that it's free software makes Custom Macros especially well-suited for researchers unfamiliar with digital techniques. 3) We believe that many researchers that avoid dedicated image analysis programs do so largely because of perceptions that such software has a steep learning curve and is generally difficult to use, is expensive and requires high-end computer hardware, and necessitates prohibitively time-consuming training for students. These concerns are not unfounded and we developed Custom Macros specifically as a solution to address these problems, which we have encountered with other software. 4) Free and open source software, like Custom Macros presents none of the issues described for proprietary software. We understand that some proprietary solutions can be used with varying degrees of transparency and our goal is certainly not to engage in a doctrinaire debate over the merits of open versus closed software code. Rather, we wish to highlight the advantages of the free and open source nature of our software.

4.4. Comparison with existing methods

4.4.1. Advantages of ImageJ

The merit of using the ImageJ platform versus competing solutions is well established and ImageJ is the most popular and widespread scientific image analysis software in use (Eliceiri et al., 2012). A comparison of ImageJ with the MetaMorph microscopy automation and image analysis software (Molecular Devices, Sunnyvale, CA) is informative. As a microscopy platform, MetaMorph controls the mechanical and light emission equipment used during imaging and provides a software interface for image acquisition. MetaMorph is also an image analysis program with object count and area measurement features similar to the core ImageJ installation (i.e., without additional plugins). In our experience, MetaMorph and core ImageJ provide similar image analysis functionality and either program can be used to achieve the same end. However, there are some key differences: 1) A great degree of ImageJ's success can be attributed to its extraordinary extensibility. User-developed plugins have expanded ImageJ's core capabilities with highly sophisticated and diverse functionality. In contrast, Metamorph is not extensible. 2) MetaMorph is proprietary software costing thousands of dollars (Austin Blanco and Pariksheet Nanda, 2012). Each installation must be licensed and is copy-protected with a USB dongle. ImageJ is free and available for download from the NIH website. 3) MetaMorph has limited operating system (OS) compatibility and, depending on the release, only works on certain versions of Windows (Molecular Devices, 2018). Users owning a MetaMorph release that runs only under Windows XP either have to pay to upgrade to a newer version or maintain a dedicated workstation with an OS that is no longer supported by Microsoft. Furthermore, MetaMorph does not run on Linux or Apple OS. ImageJ's operating system support is practically universal (see Section 2.2). 4) MetaMorph has its own macro system in which 'journals' can be recorded either in real-time or coded manually, as in ImageJ. However, unlike the ImageJ macro language, which is a powerful, complete scripting language, the MetaMorph journaling language is limited to manipulating software commands with simple control statements. 5) ImageJ is arguably one of most thoroughly well-documented scientific programs in existence. Moreover, its large and active user-base answers questions and provides technical help in online forums and communities. MetaMorph's documentation is limited to a handful of manuals and tutorials and there is no online community of users.

Please note that our goal is not to single out or disparage MetaMorph, which is a competent and highly usable program. The same criticisms apply to other proprietary image analysis solutions, such as Image-Pro Plus (Media Cybernetics, 2010, 2018a, 2018b).

4.4.2. Advantages of Custom Macros

Table 1 provides an overview of the procedures Custom Macros can run via a single keyboard shortcut or interface button click. Pressing shift, alt, control, or space while invoking some of these commands adds alternative options (see parenthetical section numbers in Table 1 for the respective Results sections detailing alternative functionality). The methodological innovation of our software's design is the integration of a diverse array of intuitive user-initiated single command macro procedures into a unified and standardized plugin suite framework. No ImageJ plugin or standalone program exists that provides the variety of user-assisted and automated measurement and processing procedures of Custom Macros nor does so in as straightforward and user-accessible of a manner. Nor does any program provide as thorough a process for automated and reproducible data generation. Custom Macros' novel cloud networking features offer a simple and decentralized solution to software administration and data storage.

Since our software utilizes ImageJ's built in functionality, some

(though not all) of the analysis procedures that are semi-automated by Custom Macros can instead be performed manually in a step-by-step fashion using the ImageJ interface. In practice, however, even a relatively simple procedure would involve numerous time-consuming steps to complete, as well as necessitating a dedicated written protocol and user training. Moreover, each new situation (e.g., color channel selection, region type) would require changes to that protocol. Ultimately, a manual approach to software interface interaction presents the following options: 1) Researcher micromanagement of observer analysts and image analysis experiments. 2) Acceptance of the potential for varying degrees of user error without a clear means of oversight. 3) Limiting the scope of an experiment's design to accommodate these logistical considerations (i.e., gathering less data). The goal of Custom Macros is to prevent the need for such difficult choices. Table 2 demonstrates the time-saving and versatile nature of using our method compared with an ad-hoc, manual approach. Note that, for the practical purpose of explanatory brevity, the macros chosen for comparison in Table 2 were selected for the relatively concise number manual of steps needed. As a contrast, the 'Completion of Measurement Analysis' macro (Section 3.3.2.2) would require dozens of manual steps, laborious copy-and-pasting, and the use of external spreadsheet or word-processing software. Our software accomplishes the procedure shown in Fig. 3D1-3 using four core Area Measurment keyboard shortcuts ('F5', 'F6', 'F7', 'F8'), one assistance shortcut ('3') and guided user interaction, as well as several optional shortcuts for user convenience (see Section 3.3.2). Replicating the same procedure with the accompanying data generation and file annotation manually for that single image would be virtually impossible, in any practical sense. Many neurohistology research laboratories perform a variety of types of analyses in their work (either in the same study or over the course of several projects), require large datasets spanning many neuroanatomical regions, and do not have a staff of dedicated bio-imaging specialists. One of the central goals in developing our software was to provide a practical solution to meet such needs.

4.5. Incorporation of third-party plugins into Custom Macros code

Custom Macros includes the implementation of third-party plugins (i.e., plugins that we have not developed but that are run by our code). Such plugins are typically found in the */ImageJ/plugins/* folder and, unless they are included in the core ImageJ or FIJI distribution, need to be downloaded from the developer's website, since they are not distributed with Custom Macros. Two third-party plugins currently utilized in Custom Macros are 'Extended Depth of Field' (Section 3.5.2.4) and 'Image Stitching' (Section 3.5.3). Publication citation information and DOI (Digital Object Identifier) links are shown in the macro interfaces. End-users can augment our software with third-party plugins by running that plugin while the Macro Recorder utility is open (menu command: Plugins- > Macros- > Record…) and pasting the output line into Custom Macros code. Alternatively, user-customized third-party plugin functionality could be incorporated into the Additional Macros menu (Section 3.5.1).

4.6. Microscopy approach applicability

All the micrograph examples we have provided were obtained through immunofluorescence microscopy. This technique segregates probe label into non-overlapping emission wavelengths, enabling the visualization of molecular targets of interest as separate 8-bit channels in RGB images. Our plugin is ideal for analyzing such images, since quantification and feature extraction of 8-bit grayscale micrographs generates immediately interpretable data.

An alternative technique for the acquiring of histological micrographs is using brightfield microscopy of tissue stained with chromogen reactants, for example DAB (3,3′-diaminobenzedine). Label in such images is not necessarily separable into meaningful 8-bit color channels and performing advanced color-based quantitative analysis requires color deconvolution techniques (Ruifrok and Johnston, 2001). Even without color deconvolution, Custom Macros-assisted analysis of raw color brightfield images is a viable option for a variety of experimental goals. The Assisted Count with Find Maxima macro (Section 3.3.1.1) yielded accurate identification and counts of cells in our tests of DAB-stained tissue. Manual counting is, of course, an alternative option (Section 3.3.1.2) and area measurement can likewise be performed if cross-sectional area and coordinate data, rather than pixel intensity, are of interest. However, since color deconvolution plugins are readily available for ImageJ ("ImageJ1 Plugins," 2018), there is no reason why appropriately color-separated brightfield micrographs cannot be treated and measured by Custom Macros in the same manner as fluorescently labeled tissue (Section 3.3.2).

4.7. Performance

A potential concern in developing plugins using macros instead of a software API or a stand-alone scripting language is program performance, specifically interface responsiveness and task run time. In the context of the structure and goals of our plugin, however, such concerns are almost certainly inconsequential. We are confident that there are no performance issues of note when using Custom Macros and have not experienced lag, memory issues, crashes, or anything else that would warrant questioning the suitability of using the ImageJ macro language in the extensive capacity that we have for plugin development. Regarding network security (see Section 3.1.3 explanation of optional networking functinality), while we would argue that the use of trusted cloud-based file hosts is generally a safer alternative to other networking approaches, users should obviously practice standard digital security diligence (i.e., firewall, anti-virus, etc.) whenever any kind of network client software is installed, regardless of whether Custom Macros is used.

4.8. Future development

Custom Macros continues to be under active development, and we are keenly interested in user feedback, feature suggestions, and collaboration. Two near-term development goals for Custom Macros are: (1) The incorporation of greater stereological functionality, such as unbiased probe overlays and accompanying sampling calculators and (2) The expansion of current statistical testing capabilities.

Conflict of interest

The authors declare that they have no conflict of interest.

Acknowledgements

Funding was supported by NSF IOS 1456743 to P.M.F. We thank our students and colleagues for their feedback and suggestions.

Appendix A. Supplementary data

Supplementary material related to this article can be found, in the online version, at doi:https://doi.org/10.1016/j.jneumeth.2019.04.009.

References

Blanco, Austin, Nanda, Pariksheet, 2012. Comparison with Other Microscopy Software - Micro-Manager [WWW Document]. Comparison with other microscopy software - Micro-Manager.. https://micro-manager.org/wiki/Comparison_with_other_microscopy_software.

Brigitte, Forster, Van De Ville, Dimitri, Berent, Jesse, Sage, Daniel, Unser, Michael, 2004. Complex wavelets for extended depth-of-field: a new method for the fusion of multichannel microscopy images. Microsc. Res. Tech. 65, 33–42. https://doi.org/10.

1002/jemt.20092.
Cromey, D.W., 2012. Digital images are data: and should be treated as such. Cell Imaging Techniques, Methods in Molecular Biology. Humana Press, Totowa, NJ, pp. 1–27. https://doi.org/10.1007/978-1-62703-056-4_1.
Molecular Devices, 2018. MetaMorph Microsoft Windows Operating System Compatibility [WWW Document]. MetaMorph Microsoft Windows Operating System Compatibility. URL. http://mdc.custhelp.com/app/answers/detail/a_id/19337.
Doube, M., Kłosowski, M.M., Arganda-Carreras, I., Cordelières, F.P., Dougherty, R.P., Jackson, J.S., Schmid, B., Hutchinson, J.R., Shefelbine, S.J., 2010. BoneJ: free and extensible bone image analysis in ImageJ. Bone 47, 1076–1079. https://doi.org/10.1016/j.bone.2010.08.023.
Ekström, P., Östholm, T., Ebbesson, S.O.E., 1992. A morphometric study of age-related changes in serotonin-immunoreactive cell groups in the brain of the coho salmon, Oncorhynchus kisutch walbaum. Exp. Neurol. 116, 204–209. https://doi.org/10.1016/0014-4886(92)90169-Q.
Eliceiri, K.W., Berthold, M.R., Goldberg, I.G., Ibáñez, L., Manjunath, B.S., Martone, M.E., Murphy, R.F., Peng, H., Plant, A.L., Roysam, B., Stuurman, N., Swedlow, J.R., Tomancak, P., Carpenter, A.E., 2012. Biological imaging software tools. Nat. Methods 9, 697–710. https://doi.org/10.1038/nmeth.2084.
Ferreira, T., Rasband, W., 2012. ImageJ1 User Guide [WWW Document]. ImageJ User Guide - IJ 1.46r. URL http://imagej.net/docs/guide/ (Accessed June 2018). .
Forlano, P.M., Licorish, R.R., Ghahramani, Z.N., Timothy, M., Ferrari, M., Palmer, W.C., Sisneros, J.A., 2017. Attention and motivated response to simulated male advertisement call activates forebrain dopaminergic and social decision-making network nuclei in female midshipman fish. Integr. Comp. Biol. 57. https://doi.org/10.1093/icb/icx053.
Geuna, S., 2000. Appreciating the difference between design-based and model-based sampling strategies in quantitative morphology of the nervous system. J. Comp. Neurol. 427, 333–339. https://doi.org/10.1002/1096-9861(20001120)427:3<333::AID-CNE1>3.0.CO;2-T.
Geuna, S., Herrera-Rincon, C., 2015. Update on stereology for light microscopy. Cell Tissue Res. 360, 5–12. https://doi.org/10.1007/s00441-015-2143-6.
Ghahramani, Z.N., Timothy, M., Kaur, G., Gorbonosov, M., Chernenko, A., Forlano, P.M., 2015. Catecholaminergic fiber innervation of the vocal motor system is intrasexually dimorphic in a teleost with alternative reproductive tactics. Brain Behav. Evol. 86, 131–144. https://doi.org/10.1159/000438720.
Ghahramani, Z.N., Timothy, M., Varughese, J., Sisneros, J.A., Forlano, P.M., 2018. Dopaminergic neurons are preferentially responsive to advertisement calls and co-active with social behavior network nuclei in sneaker male midshipman fish. Brain Res. 1701, 177–188. https://doi.org/10.1016/j.brainres.2018.09.014.
Goodman, S.N., Fanelli, D., Ioannidis, J.P.A., 2016. What does research reproducibility mean? Sci. Transl. Med. 8https://doi.org/10.1126/scitranslmed.aaf5027. 341ps12-341ps12.
Horn, Â.C.M., Rasia-Filho, A.A., 2018. The cytoarchitecture of the telencephalon of *Betta splendens* regan 1910 (Perciformes: Anabantoidei) with a stereological approach to the supracommissural and postcommissural nuclei. Anat. Rec. 301, 88–110. https://doi.org/10.1002/ar.23699.
Hosseini-Sharifabad, M., Nyengaard, J.R., 2007. Design-based estimation of neuronal number and individual neuronal volume in the rat hippocampus. J. Neurosci. Methods 162, 206–214. https://doi.org/10.1016/j.jneumeth.2007.01.009.
ImageJ1 Download [WWW Document] 2018. ImageJ1 Download. URL http://imagej.net/download.html (Accessed June 2018).
ImageJ1 Plugins [WWW Document] 2018. ImageJ1 Plugins. URL http://imagej.net/plugins/ (Accessed June 2018).
Ince, D.C., Hatton, L., Graham-Cumming, J., 2012. The case for open computer programs. Nature 482, 485–488. https://doi.org/10.1038/nature10836.
Jensen, E.C., 2013. Quantitative analysis of histological staining and fluorescence using ImageJ. Anat. Rec. 296, 378–381. https://doi.org/10.1002/ar.22641.
Media Cybernetics, 2010. Image-Pro Plus Reference Guide [WWW Document]. URL http://www.mediacy.com/images/ipplus/IPPReference.pdf.
Media Cybernetics, 2018a. Image-Pro Plus Support Center [WWW Document]. Image-Pro Plus Support Center. URL http://www.mediacy.com/support/imageproplus.
Media Cybernetics, 2018b. Image-Pro Plus Automation with Macros [WWW Document]. Image-Pro Plus Automation with Macros — Media Cybernetics Image Analysis Forums. URL http://forums.mediacy.com/categories/image-pro-plus-automation-%28macros-apps-reports%29.
Mohr, R.A., Chang, Y., Bhandiwad, A.A., Forlano, P.M., Sisneros, J.A., 2018. Brain activation patterns in response to conspecific and heterospecific social acoustic signals in female plainfin midshipman fish, *Porichthys notatus*. Brain Behav. Evol. Evol. 91, 31–44. https://doi.org/10.1159/000487122.
Preibisch, S., Saalfeld, S., Tomancak, P., 2009. Globally optimal stitching of tiled 3D microscopic image acquisitions. Bioinformatics 25, 1463–1465. https://doi.org/10.1093/bioinformatics/btp184.
Robbins, A., 2015. Effective Awk Programming: Universal Text Processing and Pattern Matching. O'Reilly Media, Inc.
Rosner, E., Rohmann, K.N., Bass, A.H., Chagnaud, B.P., 2018. Inhibitory and modulatory inputs to the vocal central pattern generator of a teleost fish. J. Comp. Neurol. 526, 1368–1388. https://doi.org/10.1002/cne.24411.
Ruifrok, A.C., Johnston, D.A., 2001. Quantification of histochemical staining by color deconvolution. Anal. Quant. Cytol. Histol. 23, 291–299.
Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B., Tinevez, J.-Y., White, D.J., Hartenstein, V., Eliceiri, K., Tomancak, P., Cardona, A., 2012. Fiji: an open-source platform for biological-image analysis. Nat. Methods 9, 676–682. https://doi.org/10.1038/nmeth.2019.
Schindelin, J., Rueden, C.T., Hiner, M.C., Eliceiri, K.W., 2015. The ImageJ ecosystem: an open platform for biomedical image analysis. Mol. Reprod. Dev. 82, 518–529. https://doi.org/10.1002/mrd.22489.
Schmitz, C., Hof, P.R., 2005. Design-based stereology in neuroscience. Neuroscience 130, 813–831. https://doi.org/10.1016/j.neuroscience.2004.08.050.
Schmitz, C., Eastwood, B.S., Tappan, S.J., Glaser, J.R., Peterson, D.A., Hof, P.R., 2014. Current automated 3D cell detection methods are not a suitable replacement for manual stereologic cell counting. Front. Neuroanat. 8. https://doi.org/10.3389/fnana.2014.00027.
Schneider, C.A., Rasband, W.S., Eliceiri, K.W., 2012. NIH Image to ImageJ: 25 years of image analysis. Nat Methods 9, 671–675. https://doi.org/10.1038/nmeth.2089.
Scott, M.L., 2015. Programming Language Pragmatics. Elsevier.
Sellaro, T.L., Filkins, R., Hoffman, C., Fine, J.L., Ho, J., Parwani, A.V., Pantanowitz, L., Montalto, M., 2013. Relationship between magnification and resolution in digital pathology systems. J. Pathol. Inform. 4. https://doi.org/10.4103/2153-3539.116866.
Wallace, C.T., Croix, C.M.S., Watkins, S.C., 2015. Data management and archiving in a large microscopy-and-imaging, multi-user facility: problems and solutions. Mol. Reprod. Dev 82, 630–634. https://doi.org/10.1002/mrd.22538.
Zelditch, M.L., Swiderski, D.L., Sheets, H.D., Fink, W.L., 2004. Geometric Morphometrics for Biologists: A Primer, 1 edition. Academic Press, Amsterdam. https://doi.org/10.1016/B978-0-12-778460-1.X5000-5.
Zhu, Y., Liu, F., Zou, X., Torbey, M., 2015. Comparison of unbiased estimation of neuronal number in the rat hippocampus with different staining methods. J. Neurosci. Methods 254, 73–79. https://doi.org/10.1016/j.jneumeth.2015.07.022.