Open Source Applied BioImage Informatics

By

Kevin W. Eliceiri

A dissertation submitted in partial fulfillment of

the requirements for the degree of

Doctor of Philosophy

(Applied BioImage Informatics)

at the

UNIVERSITY OF WISCONSIN-MADISON

2016

Date of final oral examination: 12/16/14

This dissertation is approved by the following members of the Final Oral Committee:

Elizabeth Meyerand, Ph.D., Professor, Biomedical Engineering
David Beebe, Ph.D., Professor, Biomedical Engineering
William Bement, Ph.D., Professor, Zoology
Paul DeLuca, Ph.D., Emeritus Professor, Medical Physics
Thomas Mackie, Ph.D., Emeritus Professor, Medical Physics
Willis Tompkins, Ph.D., Emeritus Professor, Biomedical Engineering

Acknowledgements

The work present here was performed in the collaborative environment of my laboratory, the Laboratory for Optical and Computational Instrumentation (LOCI). Over the last 16 years I have directed this laboratory and have been very fortunate to have had a amazing number of great colleagues, collaborators, students and staff work with me. Not only were they all essential to all the studies described but also all were very influential in my research direction, personal growth and professional training. I have learned something from each of these individuals I've had the great opportunity to work with and for that I'm very grateful. Special thanks to Curtis Rueden and Jayne Squirrell two of my long-term scientists who have been with me from the beginning and have served as key contributors.

I've also been fortunate to have had a great set of mentors through the years encouraging me in my science ranging from Dr. Hans Ris in my undergraduate work, Dr. Steve Barclay, Dr. Ken Todar, Dr. Kevin O'Connell and Dr. John White in my early graduate work and Drs. Beth Meyerand and Rock Mackie in my current graduate work. All taught me how to work hard at what you love doing and to inspire by example. I'm also fortunate to have had other great committee members including Drs. Paul DeLuca, Paul Ahlquist, Bill Bement, Willis Tompkins and David Beebe.

I'm also deeply indebted to the many good friends and close family who have encouraged me in all endeavors. Special thanks to my always supportive mother who brought me to the library, my wise brother Brian who brought me to the lab, and my kind brother Dennis who was always there for our adventures growing up. And most of all deep gratitude and love to my wife Jill who has reminded me of what is really important.

TABLE OF CONTENTS

ABSTRACT

Open Source Applied Image Bioinformatics

Kevin W. Eliceiri

Under the supervision of Dr. Elizabeth Meyerand
at the University of Wisconsin-Madison

Biological imaging has greatly advanced over the last fifty years with the now unprecedented ability to track biological phenomena in high resolution in physiologically relevant conditions over time and in space. As these imaging technologies mature and become main stream tools for the bench biologist there is great need for improved software tools that drive the informatics workflow of the imaging process from acquisition and image analysis to visualization and dissemination. To best meet the workflow challenges, these tools need to be freely available, open source, and transparent in their development and deployment. In particular it is clear that given the complexity, and heterogeneity of the modern image dataset, there cannot be a single software solution. Different imaging processing and visualization approaches need access not only to the data but also to each other. There needs to be compatibility not only in file import and export but interoperability in preserving and communicating what was done to the image. There is a great opportunity in achieving this interoperability, tools that can talk to each other not only enable new biological discovery but also efficiencies in sharing code and in many cases more precise workflows. This dissertation present efforts towards interoperability in the ImageJ2 consortiums. This consortiums are actively developing key software libraries like SciJava and SCIFIO that are utilized in dozens of software applications to parse and visualize biological image data, to the developmental benefit of not only of the applications but the libraries themselves. We are developing a complete, open source system for handling biomedical images,

including image acquisition, data storage, metadata (experimental data associated with an image), visualization, analysis, annotation, and database interconnectivity. We are adding support for multidimensional file structures such as fluorescence lifetime imaging and improving performance for handling large files. We are developing new approaches for analyzing the tumor microenvironment with a emphasis on measuring collagen. We are developing ImageJ2 as the next generation version of ImageJ with an emphasis on performance for large multidimensional files and interoperability. ImageJ2 is a new version of ImageJ seeking to strengthen both the software and its community. Internally, it is a total redesign of ImageJ, but it is backwards compatible with ImageJ 1.x via a "legacy layer" and features a user interface closely modeled after the original. Under the hood, ImageJ2 completely isolates the image processing logic from the graphical user interface (UI), allowing ImageJ2 plugins to be used in many contexts, including headless in the cloud or on a server such as OMERO, or from within another application such as CellProfiler. ImageJ2 has an N-dimensional data model driven by the powerful ImgLib2 library, which supports image data expressed in an extensible set of numeric and non-numeric types, and accessed from an extensible set of data sources. ImageJ2 is driven by a state-of-the-art, collaborative development process, including version control, unit testing, automated builds via a continuous integration system, a bug tracker and more. We are collaborating closely with related projects including Fiji, Bio-Formats and OMERO, and are striving to deliver a coherent software stack reusable throughout the life sciences community and beyond. The result is well-designed, community-driven software accessible to users yet powerful enough for programmers.

CHAPTER 1

Introduction

This chapter was adapted from a previous review in Nature Methods:

Kevin W. Eliceiri, Michael R. Berthold, Ilya G. Goldberg, Luis Ibáñez, B.S. Manjunath,Maryann E. Martone, Robert F. Murphy, Hanchuan Peng, Anne L. Plant, Badrinath Roysam, Nico Stuurmann, Jason R. Swedlow, Pavel Tomancak, Anne E. Carpenter. Biological Imaging Software Tools. Nature Methods.

The last twenty years have seen great advances in optical imaging with the ability to monitor biological phenomena with unprecedented resolution, specificity, dimensionality, complexity, and scale, all while maintaining viability and biological relevance. These novel imaging modalities, which are increasingly multi-parametric, rely heavily on computational approaches. In fact, in many cases the computational technology is just as important as the optics; not just for the digital capture that all systems now use, but in many cases also for visualizing and properly interpreting the data. In the quest for breakthroughs of biological significance, biologists are often confronted with the challenge of processing digital data of increasing complexity and richness which demands an informatics infrastructure with tools to collect, store, manipulate, analyze, interpret, and visualize vast amounts of imaging data in a reproducible way with the flexibility to refine aspects of their experimental and imaging techniques in a tight iterative loop.

Despite this great need, the bioimage informatics field is still a fairly nascent community compared to the more established hardware development side of the optical microscopy community. But with the increased mainstream adoption of advanced optical imaging approaches by biologists and the commitment by funding agencies to prioritize bioimage informatics, there has been a great increase in the number of bioimage informatics [1]tools over the last five years.

In this article, representative members of the bioimage informatics community have collaborated to review each computational step that biologists encounter when dealing with digital images, the challenges in that domain, and the overall status of available software for bioimage informatics, focusing on open source options. Our goal is to provide an overview of how open source imaging software can be utilized to provide an end-to-end laboratory solution from acquisition and data storage, to data analysis and data mining.

IMAGE ACQUISITION

Biological laboratories usually acquire images by measuring photon flux in parallel (using a camera) or sequentially (using a point detector and equipment that scans the area of interest). Although capturing an image from a camera into the computer is straightforward, in most cases image acquisition needs to be tightly synchronized with other computer-controllable equipment such as shutters, filter wheels, XY stages, Z-axis focus drives, and autofocus mechanisms (implemented in software or hardware). This automation is necessary in order to gather desired multiparametric information from a sample or to allow the unattended acquisition of large numbers of images in time-lapse series, z-stacks, multiple spatial locations in a large sample, or multiple samples in a large-scale experiment. Image acquisition software is therefore needed to communicate with these various components and coordinate their actions such that the hardware functions as quickly and flawlessly as possible, while permitting the researcher to easily design and execute the desired sequence of acquisition events.

Confocal microscope systems that require computer control to acquire an image are almost always bundled with control software whereas typical wide-field microscope systems are not, requiring the researcher to choose between several different independent software packages for microscope control and automation. But not all software packages support all microscopy-related hardware. Because the cost to write code to support hardware is high, the choice of which hardware a given software package will support is typically driven by commercial interests, and support for a hardware component can rarely be added by third parties. Hence, software compatibility with hardware is an essential consideration when planning a new microscope system.

Several commercial software packages combine image acquisition and analysis. These include Metamorph (Molecular Devices), SlideBook (3i), Image-Pro (MediaCybernetics) and Volocity (Perkin-Elmer). These software packages were often started in individual research labs and only later were commercialized. In addition, each of the major microscope companies have their own software packages, such as AxioVision (Zeiss), NIS-Elements (Nikon) and cellSens (Olympus).

The obvious advantage of commercial image acquisition packages is that they provide a turnkey solution to all “standard” image analysis strategies (snapping individual images, taking time-lapse series, three-dimensional (3-D) stacks at multiple XY positions, etc.) and when purchased as part of a system there is less danger they won’t be compatible with the system’s configuration.

It is virtually impossible, however, for individual researchers to substantially extend any of these software packages or make substantial custom hardware changes to the imaging system. This reduces the rate at which novel imaging techniques can be transferred to laboratories outside that of the inventor.

Researchers that have non-standard or frequently changing needs and equipment must often write their own code. Software development is facilitated by toolkit environments such as LabVIEW (National Instruments) and MATLAB (Mathworks), which provide interfaces to a subset of available equipment and can be used to create a graphical user interface. Developing novel imaging technologies necessitates writing instrument control code, as the needs for these novel techniques simply could not be anticipated in existing software packages. Examples of novel developments enabled by software written in research laboratories include structured illumination microscopy[2], super resolution microscopy[3-5] and Bessel beam microscopy[6]. Although these toolkit environments provide high flexibility and are well suited for tools intended for the group that wrote them, they are less appropriate for distributing these tools to others primarily because the infrastructure back-ends are costly and the distribution channels are not sufficiently developed.

Two open-source software projects, mManager and ScanImage, whose development is driven by researchers, are intended to provide tools with more flexibility than commercial tools and greater ease of use than the toolkit environments in terms of microscope control. mManager mainly targets camera-based imaging, although it is also used with scanning systems [7]. It includes an easy-to-use interface that runs as an ImageJ plugin and enables researchers to design and execute common microscopy functions as well as customized image acquisition routines. The solutions can be easily distributed as scripts or plugins. mManager's hardware abstraction layer can also be used without its ImageJ user interface in environments such as Icy, MATLAB, LabVIEW and Python, further facilitating development and transfer of software for new imaging approaches. The software framework enables any entity, academic or commercial, to write and contribute their own device adapters for hardware components. mManager provides full control of the components of the light microscope such as cameras, stages, and filter wheels. The program can be used to collect multichannel data over space and time, such as tracking fluorescently tagged cell fusion events in live cells in a multiwell plate overnight[8].

Another open source package, ScanImage, provides a software framework to control laser scanning microscopes and is used extensively for two-photon excitation microscopy [9]. It implements most standard modes of image acquisition and basic automation and supports continuous image acquisition synchronized to behavioral or physiological data, which is particularly useful for imaging in intact animals. The software framework is object-oriented and event-driven to promote extensibility, online analysis, and plugin development. ScanImage complements mManager in that it can control laser-scanning microscopes such as home built confocal systems and allow for complex recordings where high signal to noise is needed such as tracking axon signaling in neuron cultures.

These projects exemplify certain benefits that can arise organically from open source bioimaging software. Open source software in this domain aids the scientific community by facilitating the rapid dissemination of novel optical techniques from laboratories focused on instrumentation development, without requiring them to also run major software efforts. Furthermore, in the case of μManager, it now supports so many different manufacturers' cameras that its interface has essentially become a de facto standard for controlling scientific grade cameras, providing benefits to the scientific community and to commercial vendors in this arena.

IMAGE STORAGE

The vast increase in data volume and the complexity of bioimaging experiments and acquisition protocols has rapidly made a paper lab notebook an unsuitable solution for keeping track of information about imaging experiments. In fact, finding, viewing, or analyzing the data with everyday computer software is typically infeasible. Many laboratories routinely generate tens to hundreds of gigabytes of image data per day. New sophisticated automation techniques [10] promise to increase and accelerate this trend. "Enterprise-level" data generation must therefore be matched with software applications that can properly manage, view, share, process and analyze this data.

In the last ten years a number of groups have developed applications for managing large collections of scientific images. These are often referred to as 'image databases'. These image databases provide integrated platforms to organize, search, process, analyze, share and visualize data associated with biological experiments—including both images and image metadata. Such systems for image storage and retrieval are quite dependent on suitable annotation of images, both in terms of describing the experimental details that created the image as well as the automatic or human interpretation of the content of the image (see **Box 1: Image Annotation**).

The usefulness of image annotation is greatly enhanced by ontologies that formalize the names and relationships among image metadata (**Box 2: Ontologies).**

Two examples of open source bioimaging database projects are the Open Microscopy Environment's (OME) Remote Objects (OMERO) platform [11] and the Bio-Image Semantic Query User Environment (BISQUE) project [12]. Both are web-based open source projects that allow users to extend both metadata and workflow models for their individual applications. Both also invoke the idea of "remote access"-- a scientist can work with his or her data using a standard Internet connection. This type of technology is referred to as a "client-server application" where a server application holds and manages the data and delivers a view of the images, metadata, annotations, and analytic output in a client such as a web browser. This architecture provides a framework where researchers use tools hosted on a server to process and analyze large image data collections, taking advantage of the increasing power of computational resources in clusters and in the cloud.

Image databases can be searched using annotations, and relevant images can also be found by content-based image retrieval, also known as query-by-image-content. This involves searching for images that are similar to a query by some measure of distance using features or model parameters (these are discussed below under Machine Learning). As an example, the open source software OMERO.searcher provides this capability, building on the Feedback Adaptive Loop for Content-Based Retrieval (FALCON) algorithm [13] used in the Protein Subcellular Location Image Database (PSLID).

BISQUE and OMERO are just two examples of a growing ecology of software systems to help scientists manage image collections. Commercial solutions like the Columbus Image Data Management system (PerkinElmer) based on OMERO, and the PCI Image Database (Quartz), are also available. The BISQUE platform is integrated with the iPlant cyberinfrastructure to provide a scalable image management and analysis platform for plant biologists [14]. Given the vast variety of applications and contexts for bioimaging experiments, it is nearly impossible to create a universal solution for all scientists' needs. Rather, scientific image databases must be flexible and able to integrate applications and functionality demanded by the scientific application even if unanticipated by the database developers. This is why BISQUE and OMERO provide adaptable frameworks and not complete, monolithic applications. BISQUE and OMERO leverage the latest enterprise database and file server technology and couple that with the unique requirements of the imaging community. Together they represent a class of imaging database solutions that can be deployed by an individual lab, a network of labs, or publicly available repositories (see **Box 3: Public image repositories**).

This flexibility stems in part from the use of open source development practices that allow the community to review, contribute, and participate in the development project and also facilitates connections to other open-source bioimaging software. For example, both OMERO and BISQUE can work with CellProfiler and ImageJ, and implementations exist for interfacing with MATLAB. Further, BISQUE and OMERO share data models. BISQUE supports the OME XML data model and uses the OME Bio-Formats to import proprietary microscopy image data. Given the rapid development of analysis and visualization tools for imaging, this type of collaborative use of code and integration promises to provide very powerful tools for biologists now and in the near future.

IMAGE ANALYSIS & VISUALIZATION

Although there is no denying the importance of every step in the bioimaging pipeline, the heart of bioimage informatics is, of course, the images themselves and methods for their analysis and visualization. These two processes are inextricably linked and image data derived from different imaging modalities, applications and experimental designs and require a rich diversity of ever-evolving tools and techniques to extract biologically meaningful quantitative data from myriad types of microscopy images. We provide a high-level overview of the tools available and encourage readers to explore online resources and cited papers for detailed information.

Image Analysis

Biologists are increasingly interested in using image analysis to convert microscopy images into quantitative data [15, 16]. In particular, image analysis is a necessary step for experiments where hundreds or thousands of images are collected by automated microscopy, whether for screening multiple samples, collecting time or z-series data, or other technologies that generate vast volumes of image data. In addition to image analysis in a high throughout context, image processing is important for many biological studies such as quantifying the amount and localization of a signaling protein, measuring changes in structures over time, tracking invading cancer cells or looking at non-spatial data such as fluorescence lifetime data[17]. Image analysis can help ensure that results are accurate, objective and reproducible.

A vast number of image analysis algorithms and software packages have been developed for biological applications, especially in the past decade. The software packages differ in their intended application areas, the level of usability, openness of the source code, and the cost. Whereas in the past proprietary file formats often necessitated the use of only the commercial software bundled with the microscopes, third-party software programs can now be used for most images generated by most instruments, either directly or through file format readers. Given the number of open source and commercial solutions it is often difficult to choose which tool is appropriate for a given task.

Below we highlight some of the more prominent and widely used examples that have been proven useful for light microscopy across many biomedical research areas. We have focused on image analysis and visualization tools that support interoperability with each other and with software for other steps of the bioimaging workflow (such as acquisition and data storage).

Niche image analysis tools. Most image analysis software packages developed in academia are written to accomplish very specific tasks relevant to a research problem at hand. Software exists that is designed solely for particular cell types (especially neurons), particular organisms, particular assay readouts, and particular imaging modalities. There are literally hundreds of such tools both active and discontinued. One more prominent example is the "Fluorescence Association Rules for Multi-Dimensional Insight" (FARSIGHT) toolkit, which grew out of the need to map the glio-vascular substrate of brain tissue surrounding neuro-prosthetic devices. Due to the number and variety of such tools it is sometimes appropriate to find this software via web search engines or through online listings for certain biological specialties, such as the Neuroimaging Tools and Resource Clearinghouse (NITRC), which focuses on image analysis for neuroscience (http://nitrc.org) or the NIH funded Computer Integrated Systems for Microscopy and Manipulation (http://cismm.cs.unc.edu/resources/external-links/).

Generalist image analysis tools. The second category of image analysis software packages is those that can address a more general set of problems. They are typically modular and thus offer greater flexibility to multiple applications. Some commercial tools in this category include MetaMorph, Amira (Visage Imaging), Volocity, Imaris (Bitplane Scientific Software), NIS-Elements, SlideBook, ImagePro Plus (Media Cybernetics) and ZEN (Zeiss), and are often offered by microscopy companies and sold together with imaging instrumentation. There are many open-source image analysis solutions originally developed to solve the needs of a particular community but later used or expanded to other purposes, such as BioImageXD[18], Icy[19], Fiji (Fiji Is Just ImageJ)[20], Vaa3D (3-D visualization-assisted-analysis)[21], CellProfiler[22], 3D Slicer (Pieper, S., Lorensen, B., Schroeder, W. & Kikinis, R. in Proceedings of the 3rd IEEE International Symposium on Biomedical Imaging: From Nano to Macro, Vol. 1 2006), Image Slicer, Reconstruct[23], FluoRender (Wan, Y., Otsuna, H., Chien, C.-B. & Hansen, C. in Proceedings of Pacific Vis 2012 2012), ImageSurfer[24], OsiriX[25], and IMOD[26].

As with the niche image analysis software, even the generalist open source platforms are developed by researchers involved in particular biology projects and therefore although they are usable for most tasks they emphasize aspects of image analysis that are most relevant for the work they are doing themselves. For example Fiji would currently be the tool of choice in analysis of electron microscopy data, Icy offers unique features for behavioral analysis and cell segmentation and tracking, Vaa3D is heavily biased towards neurobiology and together with BioimageXD offers the best facilities for 3-D visualization (see next section). The extensive online resources available for most platforms are an excellent place to explore the strengths of the software. Similarly, the active and communicative mailing lists and chat-room channels run by the projects can be used to engage in direct interaction with the software's dedicated developers. Many innovative solutions are published and extensively cited which offers traditional means of evaluating the impact of the tools.

ImageJ (originally called NIH Image) occupies a unique position in the landscape of open source tools because it has been in use for the longest period of time and, importantly, has always been free, each of which make it the most popular and widespread multi-purpose image analysis tool[27-29]. Researchers have written hundreds of plugins and macros within this software to accomplish various image processing and analysis tasks in different application areas. One of the main reasons for the success of this tool is that scientists can leverage ImageJ's infrastructure and dissemination while focusing on developing just the application-specific algorithm at hand. This extensibility has made it a favorite among both developers and end-users and the general architecture has been later adopted by the more contemporary platforms such as Icy and Vaa3D. Due to its rich history and pioneering status ImageJ can perform a wide variety of common (and many specialized) image processing and analysis tasks, particularly within the life sciences, and the user community has grown very large. ImageJ is constantly evolving to meet the needs of the scientific community as evidenced by the community driven ImageJ2project (http://developer.imagej.net), which is developing the next generation of ImageJ to include support for a myriad of features such as large multidimensional image support, a more flexible data model, and improved developer resources.

One challenge in extensible, interoperable, community-driven software projects is the proliferation of features, options, plugins, and macros. In some contexts, having too many options is just as difficult as having too few, because selecting among them can be overwhelming. A major challenge for the scientist is not only picking which tool to use but within a tool that offers many solutions, where to begin. To address this challenge, the Fiji ImageJ distribution was developed to offer ImageJ bundled together with plugins and features specifically tailored to the microscopy community and offering new functionality for microscopy analysis[20]. The plugins are distributed through an integrated updater system that facilitates rapid feedback between plugin users and authors. Additionally, Fiji bundles several scripting languages that can be used in combination with algorithmic libraries written in Java such as ImgLib (see next section), to rapidly prototype new algorithms and facilitate productive interdisciplinary collaboration between the biology and computer science research communities.

CellProfiler is a flexible multipurpose open source image analysis tool for the life sciences that has a relatively long history and track record of utility and success[22]. It contains highly curated modules that can be mixed and matched to create customized image analysis pipelines for a variety of biological systems including cells, colonies, and *C. elegans.* Designed to accommodate high-throughput analysis, it is used to address a number of application areas including intensity and morphology measurements, phenotype scoring by machine learning, and object tracking.

More recently, comprehensive image analysis tools focusing on 3-D or high-dimensional image data have emerged. BioImageXD[18] and Icy[19] both based on the Visualization Toolkit (VTK) and the Insight Toolkit (ITK) offer many options for two-dimensional (2-D) and 3-D analysis and are extensible via plugins and macros. BioImageXD offers generation of immersive visualizations by recording fly paths through 3-D renderings of multidimensional image datasets[18]. Icy is the youngest project in the open source bioimage analysis field and as such aims to combine the very best features of the existing tools as documented by the integrated pipeline building tools, interactions with microscopy hardware and seamless update system. Vaa3D, provides state-of-art algorithms for registration, visualization and analysis of large-scale multi-dimensional imagery of complex biological systems (embryos and brains) to anatomical atlases for further integrative analysis[21].

Extensibility, interoperability and code sharing. Image analysis software is ideally extensible and interoperable, as no one tool can offer every function a researcher needs. For example, in the case of time-lapse cell analysis, a researcher may need to run a myriad of analysis routines including image denoising and deconvolution, cell identification and tracking, and measurement using advanced machine-learning analysis. The biological community has benefited from many recently constructed interfaces between image analysis software packages. It is now possible, for example, to use ImageJ for image processing and CellProfiler for cell tracking due to a link between them, enabling a more automated workflow[30]. Fiji is closely collaborating with the ImageJ2 project to provide a next generation ImageJ with improved low and high-level functionality. The benefits of this coupling are numerous, particularly in the areas of performance, multidimensional support and modularity. The current data model of ImageJ is largely limited to 2-D and 3-D making multidimensional analysis difficult. The n-dimensional model of ImageJ2 and Fiji will better support multidimensional image analysis including new modalities such as fluorescence lifetime microscopy in ImageJ2 or selective plane illumination microscopy (SPIM) uniquely available in Fiji[31] . Similarly, improvements in the ImageJ2 core will allow for code that is more centralized, easier to repackage and share with other packages in Fiji and other platforms using Java. It is encouraging to note that all the open source platforms discussed in this section are explicitly developing ways to share data and code as only interoperability will enable biologists to mix and match the best features of each platform to solve the daunting image analysis challenges.

Choosing an image analysis platform. All these options for image analysis and image informatics in general bring up a key challenge for the user: how to even choose which tool to use. Both within commercial and open source software and across these categories there is a daunting array of options with substantial feature overlap. With some aspects of image informatics workflow other than image analysis, this decision is much easier and can be based on feature sets alone as often only one tool will have the features a user would want for data acquisition or storage. However in image analysis choosing between tools often comes down to preference for familiar interface, ease of use and other intangible criteria much in the same way a computer user chooses an operating system. The developer community fully realizes this challenge, which is why there is a big movement to develop software applications to make the choice easier or unnecessary. All the open source software applications described above utilize code from other applications in the form of libraries (see below) and in some cases one platform can even be run inside another. For example all of the programs discussed use Bio-Formats to open (and in some cases save) their data and many, such as Icy and CellProfiler, directly call ImageJ plugins. It is however not possible to run Icy inside ImageJ. It is important that the projects develop their software architecture in a way that makes integration mutual as discussed in the libraries and toolkits section below. Moving beyond a typical monolithic approach means the choice for a user is easier if they don't have to decide between two alternate software packages; they can use both. Workflow systems (discussed below) that are just beginning to emerge in the biological image analysis community offer even more flexibility. They enable calling each application as components in the analysis pipeline allowing users to build their own virtual systems picking feature sets from any applications.

Image visualization

Although visualization of most two-dimensional microscopy images is trivial using a variety of software packages, modern microscopy methods enable direct capture of n-dimensional data (several channels with data across three spatial dimensions and time). The spatial dimensions alone can be very large as the addition of a computer-controlled stage allows the researcher to perform step-and-repeat microscopy, imaging large regions of tissue (millimeters to centimeters) as a montage with sub-micron resolution [32, 33]. Techniques such as serial section microscopy enable this montaging to extend to the axial dimension[34]. In SPIM the dimensionality expansion becomes multiplied as multichannel, multi-view datasets are recorded over time for observing dynamics of protein expression and localization in a live developing embryo. Below we review some of the visualization approaches currently used in microscopy. We also direct the reader to the prior Nature Methods supplement on visualizing biological data, which included an article specifically devoted to the visualization of image data[35].

Multi-dimensional imaging techniques allow direct comparison of labeled biological entities in full spatial and temporal context avoiding piecing together separate observations. However, the benefits come at the cost of more complex data storage, visualization, and analysis needs. With modern software tools and contemporary desktop graphics hardware, a three-dimensional multi-channel image cube can be projected onto the screen at interactive speeds, allowing the user to examine the data from any chosen angle and zoom factor. The speed usually results from the exploitation of hardware graphics processing engines, and careful software optimization such as utilized in the Vaa3D software [21]. When the entire image series can be loaded in computer memory, Vaa3D can be used to produce real-time 5D rendering. An alternative strategy employed, for example, by ImageJ's View5D uses cross sectional views that display only the data currently viewed at the appropriate scale. This strategy can also be easily realized on the web allowing "Google Maps" style browsing through massive multidimensional image volumes[36]. Interactive 3-D visualization of large data remains a challenge. One obvious approach is hierarchical visualization, a method that combines both global and local 3-D rendering windows in memory and additional navigation window for large image files for current desktop computers (e.g. > 8-10 Gbytes per image stack to terabytes of image datasets).

Another source of increased dimensionality is systematic imaging of large numbers of reporters in independent samples, for example, dozens of antibody-probes or thousands of 3-D registered image stacks organized as different channels. For such cases, software systems [21] now allow users to use a spread-sheet based color manager to efficiently blend and map data channels into the RGB space of computer monitors for 3-D rendering.

A whole new set of visualization needs emerge after automated image analysis, for example, segmentation (**Fig. 5a**), tracking (**Fig. 5b**), feature extraction (**Fig. 5c**), and modeling. Segmentation and tracking operations produce massive amounts of multivariate data, commonly termed features. The subsequent analysis, visualization, and interpretation of the extracted image features from multidimensional data itself requires specially adapted tools, including scatter plots, histograms, dendrograms, bi-clustering panels, database query forms, and progression displays. The FARSIGHT toolkit actively links these displays with each other and with the multidimensional image data, enabling exploration of relationships among objects and among their derived numerical data. It also enables editing of segmentation and tracking results and quantitative analysis techniques, such as the automatic identification of outliers or groups among the features. Vaa3D is able to act as a platform for further analysis of the feature spaces. For instance, a simultaneous segmentation and recognition approach built upon Vaa3D [37], which deforms the atlas of *C. elegans* [38] to best fit to a newly acquired 3-D image, allows more robust identification of single cell identities at a lower rate of segmentation error, compared to the widely used 3-D watershed-type segmentation. Vaa3D handles various image analysis tasks using a plugin interface, with which many modules such as image registration and segmentation, image input and output, and others can be glued together easily.

Multidimensional imaging is in a rapid phase of development, both from the perspective of the imaging hardware technology and the visualization and analysis software. The challenges stemming from growth in data volume to terabytes and beyond has motivated major extensions to image processing toolkits discussed in the next section.

BIOIMAGING LIBRARIES AND TOOLKITS

The image informatics community needs not only robust image processing software but also imaging libraries, or toolkits (**Fig. 6**). A library is a collection of low-level algorithms that can be used, by those with programming experience, either directly or as pieces upon which end-user software applications can be built. These toolkits serve a critical role by allowing new approaches to be rapidly and flexibly tested prior to incorporation into end-user software. They tend to be modular and enable adding functionality to end-user applications.

Although some commercial image-oriented libraries exist such as the Image Processing toolbox within MATLAB, the majority of bioimaging libraries are free and open source. Open source code is particularly helpful for libraries because of their potential to iterate and evolve rapidly with input from the scientific community.

A variety of image analysis libraries exist, differing in their programming language compatibility, memory requirements, types of images supported, algorithmic focus, speed, and level of computer science background required (**Table 2**). When evaluating in practice whether a particular library or toolkit provides the functionalities needed for a given image processing workflow, it is convenient to start by visiting the introductory tutorials that most of these software projects host. They typically showcase the common uses of the library, and can provide an overview of the library's capabilities. Online forums and mailing lists are another valuable resource. They tend to be quite active and are an effective mechanism for getting feedback on the capabilities of the software, practical advice on how to apply it to specific problems, and in some cases even ideas or assistance in modifying the source code to support new functionality. In addition to those described in more detail, below, some examples of actively used libraries include VIGRA (Vision with Generic Algorithms) (http://hci.iwr.uni-heidelberg.de/vigra/), a computer vision library with a emphasis on customizable algorithms and data structures, and EBImage[39], a modular package that leverages the R environment (http://www.r-project.org/) to segment cells and extract quantitative cellular descriptors.

Among open source bioimaging libraries, VTK and ITK play a prominent role. Both VTK and ITK are designed as a collection of data processing units, filters that take input data and produce output data. These filters can be combined into processing pipelines that provide the flexibility and adaptability required by unexpected processing needs. Both ITK and VTK are written in C++, and then are wrapped into other languages, in particular Python, Tcl and Java. These two toolkits provide support for managing datasets that are too large to fit in the main computer memory by partitioning the input data into smaller segments that are then processed one by one. VTK's main focus is on visualizing 2-D and 3-D images and geometrical meshes with a large variety of rendering techniques, together with 3-D widgets that facilitate user interactions with the objects being visualized. It also provides a collection of methods for performing information visualization, such as charts, plots, trees, and clusters. ITK is a complementary library focused on actual data processing rather than visualization; it is common to find ITK and VTK being used together by application developers. ITK provides one of the largest collections of image analysis algorithms, in particular for image segmentation, image registration, image stitching, and feature extraction. The toolkit supports N-dimensional images, with particular emphasis in 2-D, 3-D and 4-D. ITK also provides support for a large variety of image file formats, including JPEG 2000, HDF5, and TIFF images larger than 4 Gigabytes. Among the many bioimaging applications that are based on ITK or VTK are: Icy, BioImageXD, Go-Figure (http://gofigure2.org), Vaa3D, and FARSIGHT. When dealing with very large datasets that demand the use of distributed parallel computation platforms, such as clusters and supercomputers, it is common as well to take advantage of ParaView (http://paraview.org), an open source application built on top of VTK, which provides a client-server architecture.

OpenCV (http://opencv.willowgarage.com/wiki) is an open source library that provides a rich set of image analysis algorithms in the domain of computer vision, for native languages (C++, C, Python). OpenCV offers, for example, feature extraction algorithms that can identify notable structures from images, feature matching and tracking algorithms that can follow moving objects in video sequences, and calibration algorithms for correlating objects from 3-D space with features that they project into the 2-D plane of an imaging sensor. It has been used especially in the automated monitoring of phenotype behavior of animal models, which often requires analysis of hundreds of hours of video.

Because many of the widely used generalist platforms for bioimage analysis use or are written in Java, a Java library called ImgLib was developed, primarily under the Fiji project. ImgLib [20] enables the software developer to concentrate on the essence of the image analysis algorithm, transforming a mathematical formulation into a single piece of code that will run on images of any dimensionality (1-D, 2-D, 3-D... n-D), type (8bit, 12bit, 16bit, or complex types) or storage strategy (memory, disc, or internet server). Importantly, several major Java-based software projects (Fiji, ImageJ2, Konstanz Information Miner (KNIME), OMERO and Icy) are in the process of migrating towards using ImgLib as the primary or at least alternative means of image data representation, which will in the future substantially improve their interoperability.

Indeed, using a common library for representing image data enables easily moving algorithmic solutions between platforms or running different parts of the processing pipeline using different software on the same data. The issue of interoperability is crucial for long-term progress in bioimage informatics and general software libraries are one mechanism to achieve maximum integration among diverse platforms geared towards particular use cases (for example, software designed for automated processing of hundreds of thousands of images in high-throughput screens, like CellProfiler, vs. software that emphasizes processing very large images from multi-dimensional microscopy, like Fiji). As well, many bioimaging libraries have recently been linked by bridges, for example between OpenCV and ITK, and between VTK and ITK. This empowers application developers to build upon the functionalities provided by both libraries.

Another way to achieve interoperability even among platforms that rely on fundamentally incompatible programming languages (Java versus C) is to agree on common formats to store the image data and the results of computational analysis so that one software output can be seamlessly used as input for another software package. The Bio-Formats library enables usage of diverse software packages by enabling each one of them to import any proprietary format and to rely on the standardized OME-TIFF format for data exchange. As bioimage data become increasingly complex, there is a need for further development of agreed-upon data structures capable of efficiently capturing the analysis results and presenting them for downstream analysis. Integrative platforms for image data management such as OMERO or BISQUE or workflow tools such as KNIME will be indispensable for creating diverse ecosystems of cooperating bioimage analysis tools.

MACHINE LEARNING

Machine learning has been powerfully applied to experiments involving microscope images [40, 41], and is typically defined as a field concerned with creating programs whose performance improves with experience. Although non-experts may be intimidated by the concepts of machine learning, in reality these tools often require less expertise than tools that use manually configured algorithms (**Fig. 7**).

Machine learning in bioimaging is mainly used for classification – classification of either individual pixels, regions of interest in images (e.g. cells), or whole images. Whereas model-based image processing algorithms are often used to directly identify regions of interest (for example, nucleus, cytoplasm, and background), machine learning can be an alternative approach to automatically classify pixels as belonging to each class, particularly in challenging cases. Ilastik (http://www.ilastik.org/) is one open-source tool that enables researchers to train a machine learning algorithm to identify which pixels of an image belong to which class of interest, based on the researcher providing example regions of each. At the image level (whether a whole field of view or a portion thereof showing a biological entity, such as a cell), biologists often need to decide to which class a particular image belongs, such as whether a protein is in one organelle or another, or whether a cell has undergone differentiation or transformation. The classes can be defined based on the biologist's expertise or manual inspection, but they can also be defined using other information, such as which protein was tagged or what experimental manipulations were done.

To address many such problems, an experimenter provides two or more collections of images and specifies quantities (called features) that are thought or expected to be relevant to the problem (such as the number, size or shape of cell nuclei). Image processing methods are then used to calculate the features and machine learning methods are used to decide the values of the features that distinguish between the classes. However, it is often the case that the quantities to measure are unclear *a priori*, are difficult for an experimenter to define sufficiently to get robust measurements, or do not in practice achieve the desired goal. In these cases, an alternative is to extract a large set of features and let the computational methods determine, or "learn", an optimal set to use. In either case, the output is a trained classifier that can be used to sort new images into the pre-defined classes, as well as a set of statistics about the classification accuracy the system achieved. It has been demonstrated that machine learning methods can outperform human vision at recognizing patterns in microscopy images [42],[43, 44].

The features calculated as the basis for applying machine learning algorithms to images can be derived by first identifying structures of biological interest (for example, nuclei and cell boundaries), but they can also report on properties of the image as a whole like textures, pixel statistics, and factors in polynomial equations that approximate the image. Features can also be calculated on transforms of the image. The variety of image content these features represent directly determines the types of image changes to which the machine learning program will be sensitive.

Three general categories of tasks can be carried out using the features: statistical comparisons, supervised learning, and unsupervised learning. Statistical comparisons operate on the feature distributions directly, such as to determine whether two sets of images are statistically distinguishable or which image is most representative of a set, for example, to choose an image for publication, as can be done using the typical image chooser functionality in PSLID.

In supervised learning, the biologist defines relationships between images, such as by grouping together example images for different classes (a classification problem), or specifying the concentration of a drug that each sample received (a regression problem). In both cases, the program determines automatically which of the features are informative for distinguishing the classes or estimating the extent of response. Although many machine-learning programs for microscopy rely on some image pre-processing to first identify and isolate (i.e. segment) regions of interest such as cells or nuclei, successful classification has also been achieved without segmentation [45, 46]. Thus, machine learning can be applied to biological images with little or no input other than the initial sorting of images into classes. When instead the desired output is a continuous variable, regression methods, which share many concepts with classification, are used. The training images are grouped by their known outputs to construct a regression function (a “standard curve”). Examples include dose-response curves (responses to a drug at increasing concentrations) [47] and timepoints in a time series [48].

When using supervised machine learning, it is important to ensure that the trained system can generalize to new images. Different subsets of the training images are used so that the system does not become too customized to the training images, a phenomenon known as overtraining. Problems are sometimes encountered if the training set does not adequately represent the variability present in the experimental samples (such as if all of the training images are acquired on one day and there is variation from day to day). Elimination of selection bias and systematic bias in the training set is in the hands of the experimenter, and is crucial in machine learning because of the absence of an *a priori* model.

Sometimes, the classes into which images should be grouped are unknown (or uncertain). In this case, cluster analysis, a form of unsupervised machine learning, can be used to find groups within data. For example, clustering has been used to group drugs by their effects [49] and proteins by their subcellular patterns [50].

Whether to use supervised or unsupervised machine learning for a particular bioimaging problem is often a difficult choice[41]. It hinges mainly on the extent to which the user believes that all-important classes (patterns) are already known, but also on the confidence with which the user believes that the training images assigned to each class are representative of that class for all other images that will be analyzed. If labeled images are available (or can be obtained), a frequent approach is to first use supervised learning to determine how well the features can distinguish at least the major classes, and then use unsupervised learning to attempt to discover new classes, subclasses within known classes, or variation within a class over time.

Supervised and unsupervised methods can be combined, an approach referred to as semi-supervised learning. The idea is to use some information about classes, and then extend it using the data. For example, images might be grouped based on the identity of a treatment (supervised), and these samples can then be clustered into morphologically defined classes (unsupervised). An example of this used WND-CHARM (weighted neighbor distance using compound hierarchy of algorithms representing morphology) and timepoints throughout the lifespan to train a classifier to assign individual *C. elegans* a physiological age score[48].

There are also a number of approaches in which input can be introduced during the learning process. For example, in CellProfiler Analyst [51], the biologist is presented with machine classifications of new cells based on their previous input, and given the opportunity to correct errors. This new information is used to re-train the classifier. If there is a high degree of uncertainty in the classifications, a machine learner can ask the user to label selected examples in a process of active learning [41]. This process can potentially be made fully automatic, as demonstrated in [48] where a classifier is used to control an acquisition system for optimally acquiring additional examples to increase its accuracy.

Machine learning has been proven useful for a number of purposes in bioimaging, but there are some limitations. Classifiers typically do not transfer well between cell types or between different imaging systems; re-training must often be performed. A related problem is that classification and clustering systems can only “represent” the patterns they are trained on. An alternative is to try to represent a subcellular pattern or cell morphology using image-derived generative models. Given multiple images of a given pattern, a model can be trained to capture the essence and the variation in that pattern and can synthesize new images that in a statistical sense are drawn from the same distribution as the training images. This approach has only recently been applied in bioimaging; however, reasonably accurate methods for building models for nuclear and cell shape and some organelles are available.

A final limitation of machine learning approaches is that they are not typically used to handle mixtures of classes. Take for example the application of machine learning to identify to which organelle a particular protein localizes after having been trained on example images showing localization in each distinct compartment. A protein can be anywhere on a continuum between fully in one compartment to fully in one or more other compartments. This leads classifiers to produce arbitrary results when presented with images of mixed patterns, and can lead clustering algorithms to form “chained” clusters (or to artificially divide the continuum into pieces). A solution to this problem is to directly estimate how much of a given protein is in each of multiple “fundamental” compartments (e.g., 10% is in lysosomes, 90% is in the Golgi complex). This can be done given training images of proteins that are known to be in just one compartment (such as the “marker” proteins used in training classifiers), representing each image by the amount of fluorescence in distinct object types, and using various standard unmixing methods. This “supervised unmixing” approach has been successful [52] and can be done using the open source PatternUnmixer software. It is even possible to do “unsupervised unmixing” by estimating both the fundamental patterns and the mixing fractions[53].

WORKFLOW SYSTEMS

Workflow systems for data processing and analysis have begun to serve an important role across many fields in biology, including bioimaging. Workflow based tools allow researchers to flexibly and intuitively model data processing and analysis protocols and integrate a diverse array of tools without writing complex scripts or being constrained to application-focused, monolithic tools (**Fig. 8**). As such, they often address a complementary niche, where researchers are not repeating the same analysis day after day but instead are exploring and testing solutions or need to adapt to a broad array of applications. These workflow tools have become increasingly attractive because the need to process and analyze data in a sophisticated way is spreading from specialists to the majority of modern biologists. Being able to reproducibly and intuitively read, transform, process, and analyze data has become a necessary skill, very much like word processing twenty years ago.

Workflow systems can enable a field to transition from requiring that experts manually string together a series of many single-purpose software tools, to enabling non-experts to create a seamless workflow using a single integrated tool. Workflow tools have begun to serve the bioimaging community well: both expert users who know how to tweak every parameter of an algorithm, to the broader user base needing an intuitive user interface. “Visual programming” is the most common approach to user-friendliness in this domain, enabling access to a variety of functionality (e.g., image processing, data mining) while also giving access to multiple data sources (e.g., chemical, biological, textual). Further advantages are the ability to readily document and disseminate an analysis, supporting reproducible research and rapid sharing of novel approaches.

A number of workflow systems have emerged over the past decade or so. Several companies offer workflow tools, such as SAS' Enterprise Miner and IBM/SPSS Clementine, offering tools for more general data mining and statistical analysis. More focused on the life science market is Accelrys' Pipeline Pilot, which offers a combination of tools addressing chemoinformatics, image analysis, and also some data analysis capabilities. These tools are proprietary but have been offered at a steep discount for academics and are thus often used for teaching and academic research. Still, the rapid spread and continual evolution of bioimaging-related software has made it challenging for a single commercial entity to offer solutions that cover the spectrum of needs in this domain and integrate all of the relevant technology. Given the availability of state-of-the-art open source tools for individual steps of the bioimaging workflow, it is therefore no surprise that in recent years open source workflow systems have gained popularity. The workflow system itself has taken a back stage role as a data and tool integration backbone and – much like is the case for Linux – companies are more willing to invest in open standards for such critical infrastructure pieces. Open-source workflow tools such as Taverna (http://www.taverna.org.uk/) and Galaxy (http://galaxy.psu.edu/) focus on bioinformatics, while KNIME is an option that serves an even broader set of domains, including business intelligence and predictive analytics. It also has connections to other bioimaging tools, such as ImageJ, OMERO, and CellProfiler, among others, allowing the creation of complex image processing and analysis workflows. The use of workflow software in biological imaging is still very new and yet to be described in any large-scale biological research publications. However these approaches are routinely used in pharmaceutical research and have great potential to be applied to large scale, multistep analysis problems.

When considering what workflow system to use or whether a workflow approach is even necessary it's important to first define if a monolithic approach or a framework is needed. Many of the previous tools discussed in the image analysis section allow for robust workflows to be created and run, such as a series of image processing tasks in Icy or CellProfiler. The main difference between Icy and other similar applications and tools such as Taverna and KNIME is that the former are trying to do everything by themselves (sometimes incorporating other tools but not necessarily in a very flexible way). The latter are open frameworks that allow the easy integration of various other libraries and tools - in such a way that you can easily swap one tool for another one. Taverna does this by adopting a standard (web services), KNIME has its own open API and data-type format. Both are quite effective but the advantage of the open API approach is that web services are a lot harder to "archive" as web services can vary over time. In contrast if you ran a workflow in KNIME 2.4.3 in 2008 you can still run it in 2018 using that version of KNIME. When thinking about the steps involved in a workflow there are two important steps, the steps in the image analysis pipeline itself and the steps of using such image analysis in a sustainable way. For the steps in the pipeline it is important to emphasize the importance of any user being able to swap tools out easily and use a different tool or library for different tasks. The ability to visually document what one has done is also important. In thinking about the steps of using such image analysis workflows in a sustainable way, the key issues addressed by a workflow system are reproducibility, archivability and the ability to share workflows.

It is important to note that a workflow system is not needed for many image processing and analysis tasks, and many workflow needs can be addressed by the more monolithic approach in current image analysis applications. One of the strengths of a platform such as KNIME is that it is a separate piece of technology from the software pieces that actually do the work. So KNIME concentrates on the modeling of the analysis "pipeline" and allows the user to integrate whatever software libraries for the image loading, processing or analysis one wants to use (or other routines such as chemical modeling, text analysis). A user can even launch their own in-house toolbox for their preferred way to do an arbitrary image processing process.

The ability to access cutting-edge technology as different modules within a workflow system also yields a challenge: version control. The ability to reproduce an analysis precisely is critical in most scientific domains. Proprietary tools have the ability, at least in theory, to ensure that workflows continue to produce the same results with progressive updates to the software but this is not always the case. Further, those without licenses for the software cannot reproduce another researcher’s analysis, and the closed-source nature of the software limits the ability of others to rely on the software being available in perpetuity. For open-source tools, version control and reproducibility are at least feasible. This is commonly addressed by taking snapshots of the particular version of the code bases on which an end-user application depends, and storing those code bases in a combined repository, or by simply referring to a specific tagged version in the official repository of the code bases. For example, the end-user application Slicer 4.0 is built by pointing to a specific version of ITK, VTK, Python, and many other tools. Vaa3D and Icy are also built by pointing to specific versions of ITK. The multi-platform configuration tool CMake makes this process straightforward by providing the functionality of "superbuilds" and "external projects". Tools that heavily rely on external toolkits (in the extreme case, via WebService calls) allow a workflow itself to be archived, although the tools it calls may cease to function or may produce different results over time. Tools like KNIME offer users a choice in this respect by offering to call webservices or alternately to freeze a certain state, ensuring workflow reproducibility in the future.

CONCLUSIONS

As technology progresses, modern biologists must become increasingly familiar with computational techniques and software tools. Researchers using microscopy are no exception to this rule; fortunately bioimaging software has recently rapidly developed in terms of both functionality and usability. Emphasis on functionality stems from the fact that bioimaging software developers are typically embedded in, or have strong ties to, experimental biology labs. Thus the software produced by this developer community is usually in direct answer to current biology needs and often in response to new emerging problems outside the scope of commercial interest. Usability has often lagged behind, but there is increasing recognition of its value to biomedical research [54].

While the initial focus of each software project in this domain was solely to address a single step in the bioimaging workflow, the bioimaging software community has recently begun to address an important aspect of usability: interoperability. The open-source bioimaging software community has begun to communicate and collaborate, assisted and reflected by conferences such as BioImage Informatics and the Cold Spring Harbor Laboratory's Automated Imaging and

High-Throughput Phenotyping meeting. As the community gains momentum, connections among independent software projects have begun to be prioritized, and some have already been completed, as highlighted in this review. These connections greatly ease researchers' work by reducing the need to tediously transfer data between multiple software packages.

As described in commentaries in this issue [54, 55], there is a great need to not only collaborate on bioimaging ideas and approaches but also on software coding itself, including consensus on best practices and standards, software quality control, documentation, training, maintainability, and sharing of modular code. The many activities where the developers and users interact and develop their skills-for example, tutorials, conferences and hackathons-promote the progress of the software by fostering rapid innovation and interdisciplinary thinking. In fact, the community of researchers surrounding most open-source bioimaging software projects is usually much more important than the software itself. Software without community is a static resource of limited lifespan, while an active community adapts continuously to new problems. These highly networked collaborations are a common property of open-source software projects, multiplying the value of a simple software resource and enabling the attack of scientific problems too large or interdisciplinary for individual laboratories to address [56]. There are many challenges facing the imaging community, such as tracking and annotating large multidimensional datasets and retaining and sharing complex heterogeneous datasets; but also opportunities for pooling resources, such as in software coding or in testing and validation using crowd sourcing. All of these challenges necessitate the development of a robust bioimaging informatics platform.

The last twenty years have seen extraordinary biological advances driven by novel biological imaging tools, many of which directly relied on computational methods. The future of biological imaging innovation depends even more squarely on developing image informatics solutions, from acquisition to storage, analysis to mining, and visualization to dissemination. Continued advances in bioimaging computational approaches will serve not only as the foundation for new imaging methods but as the catalyst for new biological discovery that would not otherwise be possible. Resources invested in the development and maintenance of important bioimaging software applications as well as connections among them promise to yield great dividends to the thousands of biologists relying on bioimaging.

Table 1. Summary of open-source software discussed in this review

Software Name	**Primary Function**	**Website**
µManager	Microscope Acquisition	http://www.micro-manager.org
ScanImage	Microscope Acquisition	http://www.scanimage.org
OMERO	Image Database	http://www.openmicroscopy.org
Bisque	Image Database	http://www.bioimage.ucsb.org/bisque
OMERO.searcher	Image Content Search	http://murphylab.web.cmu.edu/software/searcher
Bio-Formats	Image Format Conversion	http://www.openmicroscopy.org
ImageJ	Image Analysis	http://www.imagej.nih.gov
Fiji	Image Analysis	http://www.fiji.sc
BioImageXD	Image Analysis	http://www.bioimagexd.net
Icy	Image Analysis	http://icy.bioimageanalysis.org
CellProfiler	Image Analysis	http://www.cellprofiler.org
Vaa3D	Visualization and Image Analysis	http://www.vaa3d.org
FarSight	Visualization	http://www.farsight-toolkit.org/
Visualization Tool Kit (VTK)	Bioimaging library	http://www.vtk.org
Insight Tool Kit (ITK)	Bioimaging library	http://www.itk.org
OpenCV	Bioimaging library	http://opencv.willowgarage.com/wiki/

WND-CHARM	Machine learning	http://code.google.com/p/wnd-charm/
PSLID	Machine learning	http://pslid.org
Ilastik	Machine learning	http://www.ilastik.org/
CellProfiler Analyst	Machine learning and data analysis	http://www.cellprofiler.org
PatternUnmixer	Machine learning	http://murphylab.web.cmu.edu/software
CellOrganizer	Machine learning, modeling, visualization	http://CellOrganizer.org
KNIME	Workflow system	http://www.knime.org/

Table 2: Summary of Image Analysis Libraries

Library	**Language**	**compatible with:**	**Image Dimensions**	**Computer Science Level for users**	**Algorithmic Focus**
VTK	C++	Tcl / Python / Java	2-D / 3-D	Medium	Filtering / Visualization
ITK	C++	Python / Java	n-D	Advanced	Segmentation / Registration
OpenCV	C++	Python / Java	2-D+Time	Medium	Feature Extraction / Tracking / Visualization
ImgLib	Java	Java	n-D	Advanced	Segmentation / Registration
VIGRA	C++	Python	n-D	Medium	Filtering
EBImage	R	R	2-D / 3-D	Basic	Analysis / Segmentation

Box 1: Image annotation

In order to be analyzed, retrieved, visualized, and/or shared, biological images require annotation, the process of associating images with metadata (information about the images, including information about how the images and the samples therein were created, as well as information about the content of the image itself). Whether using formal ontologies (*see Box 3: Ontologies*) or informal means of annotation, describing images in a systematic and machine-computable manner is often critical for making full use of them, either due to the scale of the bioimaging experiment or the reuse of image data for a new purpose. Unlike some other readouts, biological images can contain a vast amount of information that is often not fully extracted in the initial analysis.

Despite its recognized value, image metadata capture and annotation has not yet become widespread for bioimaging experiments [57], though efforts are underway in certain areas. A major challenge is that capturing all types of metadata for a bioimaging experiment, whether by manual or automated means, is currently time consuming and impractical. This is in part because the inherent amount of annotation required for all conceivable downstream uses of imaging data is overwhelming, causing most researchers to record just the information necessary for their own purposes. The challenge also lies in the lack of user-friendly annotation tools to entice biologists to regularly record metadata.

Some tools are in development to ease the annotation process for researchers. For image acquisition, microscopes usually automatically store information about image acquisition in the header of the resulting image files. Unfortunately, this metadata and even the pixel data itself is often stored in a proprietary form making it very difficult for other software packages to read and parse it. The Open Microscopy Environment has two solutions to help address the challenge of proprietary image formats and allow for easy and robust harvesting of acquisition metadata. The Bio-Formats project [57] is a library used by many open-source and commercial imaging software tools that allows for the full parsing of more than 120 proprietary image formats and the accurate conversion of the proprietary metadata to the OME-XML data model. The OME-TIFF project is a container format for the OME-XML data model and serves as the main export format of Bio-Formats. OME-TIFF is therefore a practical choice for software tools that wish to record their metadata in an open image format. The OMERO data system of the Open Microscopy Environment also offers image annotation with current support for text and graphical annotation of regions of interest for images stored in the database. The CCDB's Web Image Browser tool is in development to enable manual annotation of the organism type and anatomical region where the image or volume is found (http://openccdb.org/software/index.shtm#wib). To encourage biologists to record experimental details leading to the capture of images, especially for complex experiments, ProtocolNavigator captures protocols as a visual map (http://vizbi.org/Posters/2011/D06).

Although annotation of images must often be done through manual curation, approaches for automated annotation of image content are in development (see "Machine learning" section) and are especially needed given the sheer number and volume of images that are now routinely produced. These systems, whether providing automatic harvesting of metadata at acquisition or manual annotation post acquisition, are in the early stages of implementation and adoption. As the field increases its demand for quantitative analysis and robust curation and sharing of the image data, the need for full ontologies and annotations will increase. Annotation is needed at every stage of the biological imaging workflow, but systems covering both manual and automated annotation of the full workflow still need to be developed and implemented.

Box 2: Ontologies

Annotation of images, in order to be useful for image retrieval and analysis across experiments and laboratories, is greatly enhanced through the use of ontologies. Ontologies are formal expressions of human knowledge about a domain in machine readable form [58]. Ontologies enable the consistent description of image metadata, such as the type of microscopy or stains used and the cell or tissue type imaged. They can also enable consistent description of visual annotations by defining a set of classes, such as cell or organelle, and the relationships between them—an organelle is part of a cell.

There are a number of valuable benefits to image annotation with ontologies. First, at the most basic level, ontologies provide an effective and flexible means to apply controlled vocabularies to annotation. This allows multiple terms (synonyms, plurals and lexical variants) to be resolved as describing the same item and disambiguates multiple items being described using the same single term (for example, the nucleus of cell, the nucleus of the brain, and the nucleus of an atom). Second, ontologies are computable. A well-constructed ontology names each class via a unique identifier, ideally structured in the form of a uniform resource identifier (URI), which serves as an easily computable "handle". Third, when constructed using formal logical languages like OWL (Web Ontology Language), ontologies can perform categorization through similar reasoning processes to a human. For example, ontologies enable the generation of new hierarchies based on rules and axioms (for example, a Purkinje cell is a type of GABAergic neuron because it is both a type of neuron and uses GABA as a neurotransmitter). By expressing image annotation in machine computable form as a formal ontology, human knowledge can be brought to bear on effective search and interpretation of image data, especially across multiple disciplines, scales, and modalities.

Ontologies and ontology-based services that can be built into databases, and tools exist for certain metadata relevant to bioimaging experiments and are in development for others, through efforts such as the Open Biological Ontologies project, the Gene Ontology and the National Center for Biomedical Ontologies Bioportal and the Neuroscience Information Framework. These ontologies cover major biological entities, such as cell components (Gene Ontology), chemicals of biological interest (Chebi), anatomical entities (Foundational Model of Anatomy, NIFSTD) and cell types (Cell Ontology). Other ontologies cover entities describing experimental acquisition (OME-XML), experimental techniques and protocols (Ontology of Biomedical Investigation) and data elements (Information Artifact Ontology). However, the process of formal ontology construction is slow and may not keep pace with rapidly evolving imaging technologies. Systems have also been proposed by which highly granular vocabulary terms can be developed and evolved locally into an ontology by the experimentalist, with guidance provided that encourages concordance with existing ontologies ([59] and http://xpdb.nist.gov/bioroot/bioroot.pl).

Box 3: Public image repositories

Public bioimage data repositories have been available for a number of years, and the number is growing [60]. Some repositories are geared towards education and outreach, such as the The Cell: An Image Library (http://www.cellimagelibrary.org) and The Cell Centered Database (CCDB) (http://ccdb.ucsd.edu), which have merged in 2012. Others gather sets of images that accompany published papers, such as the Journal of Cell Biology (JCB) DataViewer (http://jcb-dataviewer.rupress.org), which provides full access to original multi-dimensional image data associated with articles published in JCB. Still others are more focused on providing research resources, often focused on particular domains. These include the Human Protein Atlas (http://proteinatlas.org), the Allen Brain Atlas (http://www.brain-map.org), the e-Mouse Atlas Project (http://www.emouseatlas.org), the Protein Subcellular Location Image Database (http://pslid.org), the Worm Atlas (http://www.wormatlas.org), Drosophila Image Repositories (http://flybase.org) and LOCATE (http://locate.imb.uq.edu.au). The methods for searching, analyzing and distributing images from these databases are still developing. Using standard methods of tracking access and downloads will help identify which data are most valuable for the community and which applications and functionality are most scientifically useful.

Brief Introduction to other Chapters

In this chapter (Chapter 1) we reviewed the rationale and status of a range of open source image informatics tools from several groups. This included tools for acquisition, data mining and analysis. This chapter was designed to overview the diversity of biological imaging tools and how they are working together including with the software from our group. The remaining chapters focus on image analysis software from my lab, with Chapter 2 being a description of our overall efforts in ImageJ and related tools, Chapter 3 and 4 on novel collagen analysis tools, Chapter 5 introducing ImageJ2 formally and describing its primary development, Chapter 6 reviewing efforts to add FLIM analysis to ImageJ, and Chapter 7 on Spectral Analysis. Chapter 8 summarizes some plan future areas to explore in interoperability in image informatics. While the topics are diverse in these chapters, ranging from how to create a refactored and reengineering ImageJ platform to meet the needs of the biological imaging community, specific software tools for collagen and lifetime analysis to describing a novel spectral microscope and its software, there are several points of intersect. In all cases we are trying to develop tools for analysis high resolution images and extracting relevant features in fluorescence. Some of these tools are more about creating a modular and reusable software stack that can be used by others. In other cases its about the deployments of this software to solve a particular biological problem. A central theme of all these chapters is they are bioimage informatics tools that are designed to not only be open but interoperable with other software and application uses whenever possible.

References:

1. Peng, H. Bioimage informatics: a new area of engineering biology. *Bioinformatics* **24**, 1827-1836 (2008).
2. Gustafsson, M.G. Nonlinear structured-illumination microscopy: wide-field fluorescence imaging with theoretically unlimited resolution. *Proc Natl Acad Sci U S A* **102**, 13081-13086 (2005).
3. Huang, B., Wang, W., Bates, M. & Zhuang, X. Three-dimensional super-resolution imaging by stochastic optical reconstruction microscopy. *Science* **319**, 810-813 (2008).
4. Hess, S.T., Girirajan, T.P. & Mason, M.D. Ultra-high resolution imaging by fluorescence photoactivation localization microscopy. *Biophys J* **91**, 4258-4272 (2006).
5. Jones, S.A., Shim, S.H., He, J. & Zhuang, X. Fast, three-dimensional super-resolution imaging of live cells. *Nat. Methods* **8**, 499-508 (2011).
6. Planchon, T.A. et al. Rapid three-dimensional isotropic imaging of living cells using Bessel beam plane illumination. *Nat. Methods* **8**, 417-423 (2011).
7. Edelstein, A., Amodaj, N., Hoover, K., Vale, R. & Stuurman, N. Computer control of microscopes using µManager. *Curr. Protoc. Mol. Biol.* **92**, 14.20.11-14.20.17 (2010).
8. Lin, H.P., Vincenz, C., Eliceiri, K.W., Kerppola, T.K. & Ogle, B.M. Bimolecular fluorescence complementation analysis of eukaryotic fusion products. *Biol. Cell.* **102**, 525-537 (2010).
9. Pologruto, T.A., Sabatini, B.L. & Svoboda, K. ScanImage: flexible software for operating laser scanning microscopes. *Biomed Eng Online* **2**, 13 (2003).
10. Conrad, C. et al. Micropilot: automation of fluorescence microscopy-based imaging for systems biology. *Nat. Methods* **8**, 246-249 (2011).
11. Allan, C. et al. OMERO: flexible, model-driven data management for experimental biology. *Nat. Methods* **9**, 245-253 (2012).
12. Kvilekval, K., Fedorov, D., Obara, B., Singh, A. & Manjunath, B.S. Bisque: a platform for bioimage analysis and management. *Bioinformatics* **26**, 544-552 (2010).
13. Wu, L., Faloutsos, C., Sycara, K.P. & Payne, T.R. in Proceedings of the 26th International Conference on Very Large Data Bases (Morgan Kaufmann Publishers Inc., 2000).
14. Goff, S.A. et al. The iPlant Collaborative: Cyberinfrastructure for plant biology. *Frontiers in Plant Science* **2** (2011).
15. Glory, E. & Murphy, R.F. Automated subcellular location determination and high-throughput microscopy. *Dev Cell* **12**, 7-16 (2007).
16. Ljosa, V. & Carpenter, A.E. Introduction to the quantitative analysis of two-dimensional fluorescence microscopy images for cell-based screening. *PLoS Comput Biol* **5**, e1000603 (2009).
17. Lakowicz, J.R. Principals of Fluorescence Spectroscopy. (Academic Press, New York; 1999).
18. Kankaanpää, P. et al. BioImageXD: open general purpose and high-throughput image processing platform. *Nat. Methods* (In Press).
19. de Chaumont, F. et al. Icy: an open bioimage informatics platform for extended reproducible research. *Nat. Methods* (In Press).

20. Schindelin, J. et al. Fiji - an Open Source platform for biological image analysis. *Nat. Methods* (In Press).
21. Peng, H., Ruan, Z., Long, F., Simpson, J.H. & Myers, E.W. V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets. *Nat. Biotechnol.* **28**, 348-353 (2010).
22. Carpenter, A.E. et al. CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biol* **7**, R100 (2006).
23. Fiala, J.C. Reconstruct: a free editor for serial section microscopy. *J. Microsc.* **218**, 52-61 (2005).
24. Feng, D. et al. Stepping into the third dimension. *J. Neurosci.* **27**, 12757-12760 (2007).
25. Rosset, A., Spadola, L. & Ratib, O. OsiriX: An open-source software for navigating in multidimensional DICOM images. *J. Digit. Imaging* **17**, 205-216 (2004).
26. Kremer, J.R., Mastronarde, D.N. & McIntosh, J.R. Computer visualization of three-dimensional image data using IMOD. *J. Struct. Biol.* **116**, 71-76 (1996).
27. Collins, T.J. ImageJ for microscopy. *Biotechniques* **43**, 25-30 (2007).
28. Abramoff, M., Magalhaes, P. & Ram, S. Image processing with ImageJ. *Biophotonics International* (2004).
29. Schneider, C.A., Rasband, W.S. & Eliceiri, K.W. NIH Image to ImageJ: 25 years of image analysis. *Nat. Methods* (In Press).
30. Kamentsky, L. et al. Improved structure, function and compatibility for CellProfiler: modular high-throughput image analysis software. *Bioinformatics* **27**, 1179-1180 (2011).
31. Preibisch, S., Saalfeld, S., Schindelin, J. & Tomancak, P. Software for bead-based registration of selective plane illumination microscopy data. *Nat. Methods* **7**, 418-419 (2010).
32. Tsai, C.L. et al. Robust, globally consistent and fully automatic multi-image registration and montage synthesis for 3-D multi-channel images. *J Microsc* **243**, 154-171 (2011).
33. Preibisch, S., Saalfeld, S. & Tomančák, P. Globally optimal stitching of tiled 3D microscopic image acquisitions. *Bioinformatics* **25**, 1463-1465 (2009).
34. Saalfeld, S., R, F., A, C. & P, T. Elastic Volume Reconstruction from series of ultra-thin microscopy sections. *Nat. Methods* **In Press** (2012).
35. Walter, T. et al. Visualization of image data from cells to organisms. *Nat. Methods* **7**, S26-41 (2010).
36. Saalfeld, S., Cardona, A., Hartenstein, V. & Tomančák, P. CATMAID: collaborative annotation toolkit for massive amounts of image data. *Bioinformatics* **25**, 1984-1986 (2009).
37. Qu, L. et al. Simultaneous recognition and segmentation of cells: application in *C.elegans*. *Bioinformatics* **27**, 2895-2902 (2011).
38. Long, F., Peng, H., Liu, X., Kim, S.K. & Myers, E. A 3D digital atlas of C. elegans and its application to single-cell analyses. *Nat Methods* **6**, 667-672 (2009).
39. Pau, G., Fuchs, F., Sklyar, O., Boutros, M. & Huber, W. EBImage--an R package for image processing with applications to cellular phenotypes. *Bioinformatics* **26**, 979-981 (2010).
40. Shamir, L., Delaney, J.D., Orlov, N., Eckley, D.M. & Goldberg, I.G. Pattern recognition software and techniques for biological image analysis. *PLoS Comput. Biol.* **6**, e1000974 (2010).

41. Murphy, R.F. An active role for machine learning in drug development. *Nat. Chem. Biol.* **7**, 327-330 (2011).
42. Murphy, R.F., Velliste, M. & Porreca, G. Robust numerical features for description and classification of subcellular location patterns in fluorescence microscope images. *Journal of VLSI Signal Processing* **35**, 311-321 (2003).
43. Nattkemper, T.W., Twellmann, T., Ritter, H. & Schubert, W. Human vs machine: evaluation of fluorescence micrographs. *Comput Biol Med* **33**, 31-43 (2003).
44. Johnston, J., Iser, W.B., Chow, D.K., Goldberg, I.G. & Wolkow, C.A. Quantitative image analysis reveals distinct structural transitions during aging in Caenorhabditis elegans tissues. *PLoS One* **3**, e2821 (2008).
45. Huang, K. & Murphy, R.F. From quantitative microscopy to automated image understanding. *J Biomed Opt* **9**, 893-912 (2004).
46. Shamir, L. et al. Wndchrm - an open source utility for biological image analysis. *Source Code Biol Med* **3**, 13 (2008).
47. Loo, L.H., Wu, L.F. & Altschuler, S.J. Image-based multivariate profiling of drug responses from single cells. *Nat Methods* **4**, 445-453 (2007).
48. Jackson, C., Glory-Afshar, E., Murphy, R.F. & Kovacevic, J. Model building and intelligent acquisition with application to protein subcellular location classification. *Bioinformatics* **27**, 1854-1859 (2011).
49. Perlman, Z.E. et al. Multidimensional drug profiling by automated microscopy. *Science* **306**, 1194-1198 (2004).
50. Chen, X. & Murphy, R.F. Objective clustering of proteins based on subcellular location patterns. *J Biomed Biotechnol* **2005**, 87-95 (2005).
51. Jones, T.R. et al. Scoring diverse cellular morphologies in image-based screens with iterative feedback and machine learning. *Proc Natl Acad Sci U S A* **106**, 1826-1831 (2
52. Peng, T. et al. Determining the distribution of probes between different subcellular locations through automated unmixing of subcellular patterns. *Proc Natl Acad Sci U S A* **107**, 2944-2949 (2010).
53. Coelho, L.P., Peng, T. & Murphy, R.F. Quantifying the distribution of probes between subcellular locations using unsupervised pattern unmixing. *Bioinformatics* **26**, i7-i12 (2010).
54. Carpenter, A.E., Kamentsky, L. & Eliceiri, K.W. A call for bioimaging software usability. *Nat. Methods* (In Press).
55. Cardona, A. & Tomancak, P. Current Challenges in Open Source Bioimage Informatics. *Nature Methods* **In press** (2012).
56. Nielsen, M. Reinventing Discovery: The New Era of Networked Science. (Princeton University Press, 2011).
57. Linkert, M. et al. Metadata matters: access to image data in the real world. *J. Cell. Biol.* **189**, 777-782 (2010).
58. Larson, S.D. & Martone, M.E. Ontologies for Neuroscience: What are they and What are they Good for? *Front Neurosci* **3**, 60-67 (2009).
59. Plant, A.L., Elliott, J.T. & Bhat, T.N. New concepts for building vocabulary for cell image ontologies. *BMC Bioinformatics* **12**, 487 (2011).
60. Swedlow, J.R. Finding an image in a haystack: the case for public image repositories. *Nat. Cell. Biol.* **13**, 183 (2011).

Chapter 2: ImageJ Ecosystem

Adapted from: The ImageJ ecosystem Johannes Schindelin, Curtis T. Rueden, Mark C. Hiner and Kevin W. Eliceiri. Special Issue on Imaging in Molecular Reproduction and Development.

Microscopy technology advances rapidly, enabling increasingly affordable, faster and more precise imaging, which necessitates correspondingly more advanced image processing and analysis techniques. A wide range of software is available to that end, from commercial to academic, from special-purpose to Swiss army knife, from small to large—but a key characteristic of software suitable for scientific inquiry is its accessibility. Open-source software—which can be freely inspected, modified and redistributed—inherently lends itself to scientific endeavors; in particular, the software ImageJ has had a huge impact on life sciences and continues to do so. Ever since its inception, ImageJ enjoys a growing number of users, in part because it is freely available, in part because it has fostered a vibrant and helpful community. Scientists as diverse as interested hobbyists, technical assistants, students, scientific staff and advanced researchers use ImageJ on a daily basis and exchange knowledge on the ImageJ mailing list. ImageJ's use ranges from data visualization and teaching to advanced image processing and statistical analysis. Due to its extensibility, it continues to attract not only biologists, but also more and more computer scientists who wish to effectively implement specific image processing algorithms. In this review, we will use the ImageJ project as a case study of how open-source software fosters an ecosystem of software tools, making an abundance of image analysis technology easily accessible to the scientific community, exploring what makes ImageJ so popular, how it impacts life science, how it inspires other projects and how ImageJ itself is influenced by coevolving projects.

Introduction

Ever since digital imaging equipment entered the world of science, life scientists have collaborated with computer scientists to apply image-processing techniques to analyze biomedical image data. The aim is to use computational means to accelerate repetitive tasks, but also to obtain quantitative results: statistical results are much more compelling, scientifically speaking, than qualitative observations. To this end, computer vision experts designed techniques that are applied successfully to biomedical images, and developing such methods has become a field of research of its own. Biomedical image processing is a subset of computer vision research with its own specific challenges—e.g., low-light conditions required to keep the imaged specimen alive. Compared to conveniently uniform footage from e.g. a video camera, biomedical images require substantial knowledge about the physical intricacies of the optics involved, coupled with textbook computer vision expertise, for sound image processing. The advances in biomedical image processing techniques have made it possible to corroborate research outcomes by quantifying them in a rigid, statistical manner, at the same time raising the bar for life science research when it comes to substantiating scientific observations. Therefore, the need arose to provide life scientists with accessible ways to execute image processing and analysis techniques to underpin their research.

A large number of reviews have been written on the subject of biomedical image processing and analysis tools in general (e.g. Eliceiri 2012) and on tools to perform specific tasks (e.g**.** Meijering 2006, Pham 2000). It would be beyond the scope of any review to present a top-level overview of available image processing techniques. Rather, in this review we will use the ImageJ project (Schneider 2012) as a case study of how open-source software fosters an ecosystem of software tools, making an abundance of image analysis technology easily accessible to the scientific community.

ImageJ facilitates scientific inquiry

In 1987, when Wayne Rasband—then employed at the National Institute of Health (NIH)—released ImageJ's previous incarnation, "NIH Image," it entered a scientific landscape that already saw highly advanced scientific image processing software targeting computer scientists (e.g. Cristy 1994, Konstantinides 1994). But a few aspects set ImageJ (and NIH Image) apart: application and source code are both made available free of cost, it has a very simple user interface, and it is able to run on affordable desktop machines. Additionally, as a project that welcomes source code contributions, ImageJ attracts pure users of the software and software programmers alike.

This open and inclusive development style is very much by design: it was and is Wayne Rasband's understanding that the NIH—being funded through public money—must serve the public good. As a consequence, ImageJ is open source. This makes it as scientifically relevant as possible: the program's functionality can be changed in any which way, its inner workings can be scrutinized, and it provides an excellent resource to learn how to implement image processing algorithms. Would ImageJ be offered for free, but its source code unavailable or restrictively licensed, it would essentially be a black box that hinders interested parties from understanding and validating fully what it does—a concept that is simply incompatible with the very nature of scientific research.

ImageJ's accessibility also makes it is an ideal teaching resource: it can be obtained free of cost, incurring only the usual maintenance costs for the classroom computers instead of a prohibitively expensive set of per-seat licenses. Course participants can even use their own laptops, minimizing the effort to apply what has been learnt in their daily work. In addition, macros and plugins dedicated to demonstrating image analysis concepts are easily added to the ImageJ installations. Just to name one prominent example: the Spirals macro (http://imagej.net/Spirals) is designed to dispel the all-too-common notion that it is reliable to quantify colors by eye. ImageJ is not only used to teach image processing fundamentals, though: ImageJ is open source and can be extended easily by plugins written in Java (requiring only a Java Development Kit that is available free of cost, too), and therefore became one of the most popular tools in teaching development of image processing algorithms (Burger 2010), too. For example, the Biomedical Imaging Group of the EPFL (e.g. Forster 2004, Delgado-Gonzalo 2012) offers a regular semester-long image-processing course using ImageJ to teach students how to implement image-processing algorithms.

Furthermore, the ImageJ community set an example how research can be more effective by having an interdisciplinary forum where knowledge is disseminated, where questions that might appear difficult to experts in one scientific field may be easily answered by an expert in another field. The diversity of the expertise present on ImageJ's mailing list, ranging from experimental biologists to paleontologists to astronomers to computer scientists, as well as the polite tone, give rise to insightful scientific exchanges on a daily basis.

It may be this collaborative spirit that had the biggest impact on how life science research is performed today as it enables and inspires the way other scientific projects are run. More and more scientists acknowledge the fact that software plays the same important role as other materials and methods in scientific research. Just as e.g. protocols or genetic mutant lines are expected to be shared with other scientists to ease validation and to facilitate subsequent research, source code used to produce scientific findings must be made available to other scientists. Given only the text of a scientific publication, it would be too cumbersome for an independent group to reproduce the findings had they no easy access to the same materials and methods, and in fact in that case, there would be no guarantee that the experiments could ever be reproduced faithfully. It is exactly the same for software: given only a description of an algorithm, it is not only an undue burden to ask independent groups who seek to validate the published results to re-implement the algorithm themselves, there would also be no relation whatsoever between the original and the independent implementation, rendering any attempt at a validation of the original implementation invalid.

ImageJ is the heart of a software ecosystem

The parallels between the technological landscape of an open-source software project and a biological ecosystem are numerous. The freedom to modify and redistribute open-source software leads to a proliferation of different software "species" each occupying its own specialized functional niche. Projects diverge over time (or "fork") giving rise to mutants. Components interoperate with and benefit one another in continuous co-operational evolution—or compete with each other, providing multiple solutions to similar problems.

The ImageJ project exemplifies this sort of open-source software ecosystem. Continuous advances in scientific understanding necessitate a constant state of flux in scientific software: new projects crop up, old projects are succeeded and retired, and projects collaborate to answer scientific questions and ensure their mutual longevity. This process is facilitated by a vibrant community of scientists exchanging ideas, knowledge, advice and programs on public channels such as the ImageJ mailing list. By offering the program for free, encouraging participation, and setting an example through a gentle and helpful presence, Wayne Rasband undoubtedly helped the community to flourish and to develop a culture that truly follows the scientific tradition that let Newton state: *"If I have seen further it is by standing on the shoulders of giants."* Every scientific discovery is based on other scientific research, and sharing knowledge as it is done in the ImageJ community provides fertile soil for new discoveries.

From a technical perspective, ImageJ provides a user interface with functions to load, display and save images; basic image processing functionality such as convolution filters; and an extension mechanism including support for plugins and macros. It is this **extensibility** that is truly the root of ImageJ's effectiveness: advanced image processing methods—e.g., wavelet analysis or active contour segmentation—are not provided by the core application, but rather made available as third-party **plugins** by specialists in the corresponding fields. ImageJ provides a so-called Application Programming Interface (API) for these plugins—special-purpose software components that extend ImageJ's functionality by offering additional commands available via menu entries. Plugins can tap into ImageJ's existing functionality—e.g. reading and writing images, requesting user input via dialogs, displaying histograms, plots and spreadsheets, etc.—without having to implement the same functionality from scratch again, a process often labeled as "reinventing the wheel" by software developers. Furthermore, ImageJ has the ability to record and replay **macros**—lists of simple instructions that describe exactly the actions performed by the user, and which ImageJ can interpret to repeat the exact same actions—which dramatically increases the versatility of the program. Macros enable scientists to automate their image processing workflows, to provide their colleagues with exact documentation of their workflows, and to collaborate on improving workflows, all of which with little to no formal training.

Finally, ImageJ is not only an application—it is also what software developers call a *software library*: its functionality can be used without displaying ImageJ's user interface by calling functions directly via the API from other Java programs. This makes it possible for scientists to develop special-purpose, simplified user interfaces that react to user input by calling ImageJ behind the scenes to perform the actual image processing. The API also makes it possible to perform heavy-duty computing jobs in the background, or to send them off to a cluster of computers or even a cloud service. This sort of **interoperability** between ImageJ and other software packages enables a remarkable cross-pollination throughout its ecosystem.

The ImageJ ecosystem is diverse

Thanks to the characteristics described above, ImageJ has engendered a very large ecosystem, in which the most basic organisms are the macros and plugins (see e.g. http://imagej.net/plugins/). These plugins are highly varied, capable of modifying existing functionality (e.g., support for a new file format) or introducing completely new operations (e.g., an unsupervised classification algorithm). This diversity of tools is one of ImageJ's key strengths: plugins can be freely combined to accomplish complex analyses; users can select between multiple plugins offering similar functionality, choosing the most appropriate tool for each situation; and in contrast to a monolithic approach, ImageJ's extensible structure means new functionality can always be layered on top of the existing tools, rather than needing to be created from the ground up.

Given the expanse of software projects in the ImageJ ecosystem, we will focus on the aspects that distinguish these projects from one another. The most obvious distinction between plugins is simply their functionality: within image processing, the most common plugin categories are visualization, preprocessing, segmentation, registration and tracking.

Visualization. Given the ubiquity of images as data within the life sciences, it may be surprising that visualization of said data is subtly complex. We often dangerously over-estimate our ability to visually quantitate and differentiate (see e.g. http://imagej.net/Adelsons_Squares). Nonetheless, qualitative visualization is a vital technique, with plugins such as the 3D Viewer (Schmid 2010) and Volume Viewer (http://imagej.net/Volume_Viewer) providing hardware-accelerated, interactive 3D rendering within ImageJ. Using these plugins, scientists can assemble 3-dimensional structures from a collection of images, allowing the inspection of regions in the context of their surroundings. Although these plugins may not, themselves, be quantitating pixel values for scientific benefit, the additional perspectives provided can be invaluable in identifying regions of interest.

Preprocessing. Algorithms which alter image pixels—e.g., enhancing the contrast, reducing the noise or subtracting background signal—are often required to facilitate subsequent analysis, but care is needed to avoid introducing artifacts by preprocessing the images inappropriately. For example, a general CLAHE plugin (see http://imagej.net/CLAHE) might be used to make the borders of cellular structures more distinct (see figure 1), making segmentation possible. Various types of image artifacts and noise often end up with their own optimized plugin(s); in time-based image analysis photobleaching can be a significant problem, but the Bleach Correction (http://imagej.net/Bleach_Correction) plugin provides the potential for correction.

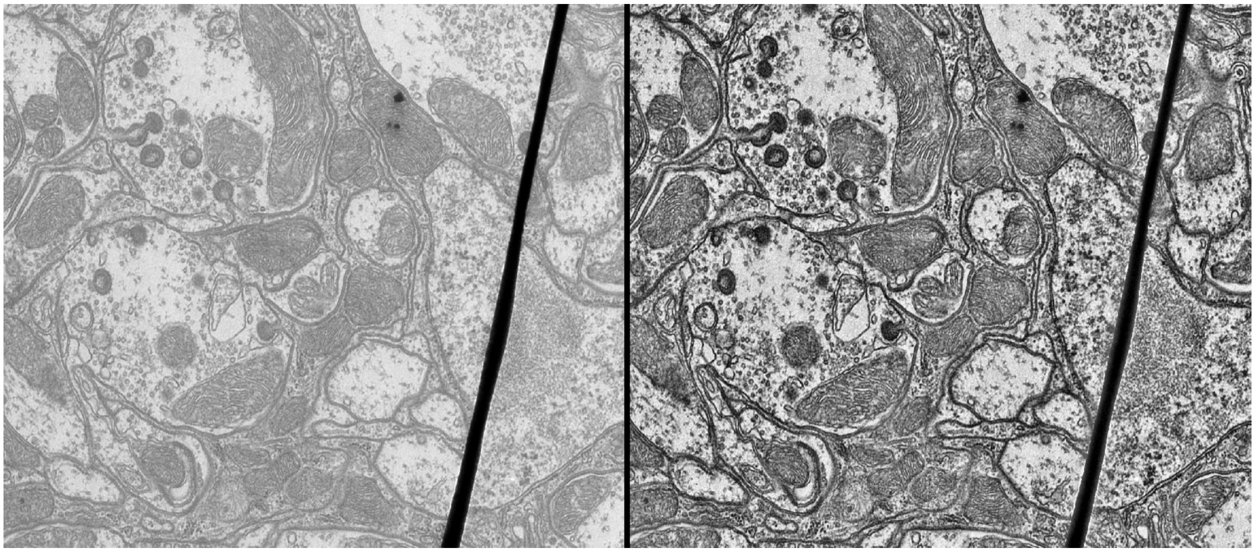


Figure 1: TEM image before and after CLAHE contrast enhancement.

Segmentation entails identifying regions or structures of interest automatically or interactively. Identifying the extents of these regions within the image allows for quantification and further analysis of the desired structures. The Trainable Weka Segmentation (TWS) plugin (Kaynig 2010) allows users to define classes of objects within an image. These classes are then applied to the original image, creating a fully segmented dataset (see figure 2). Specialized segmentation plugins such as the Simple Neurite Tracer (SNT) (Longair 2011), itself built on the 3D Viewer, can provide a more automated experience by leveraging pre-trained models (identifying tube-like structures in 3D stacks, in the case of the SNT).

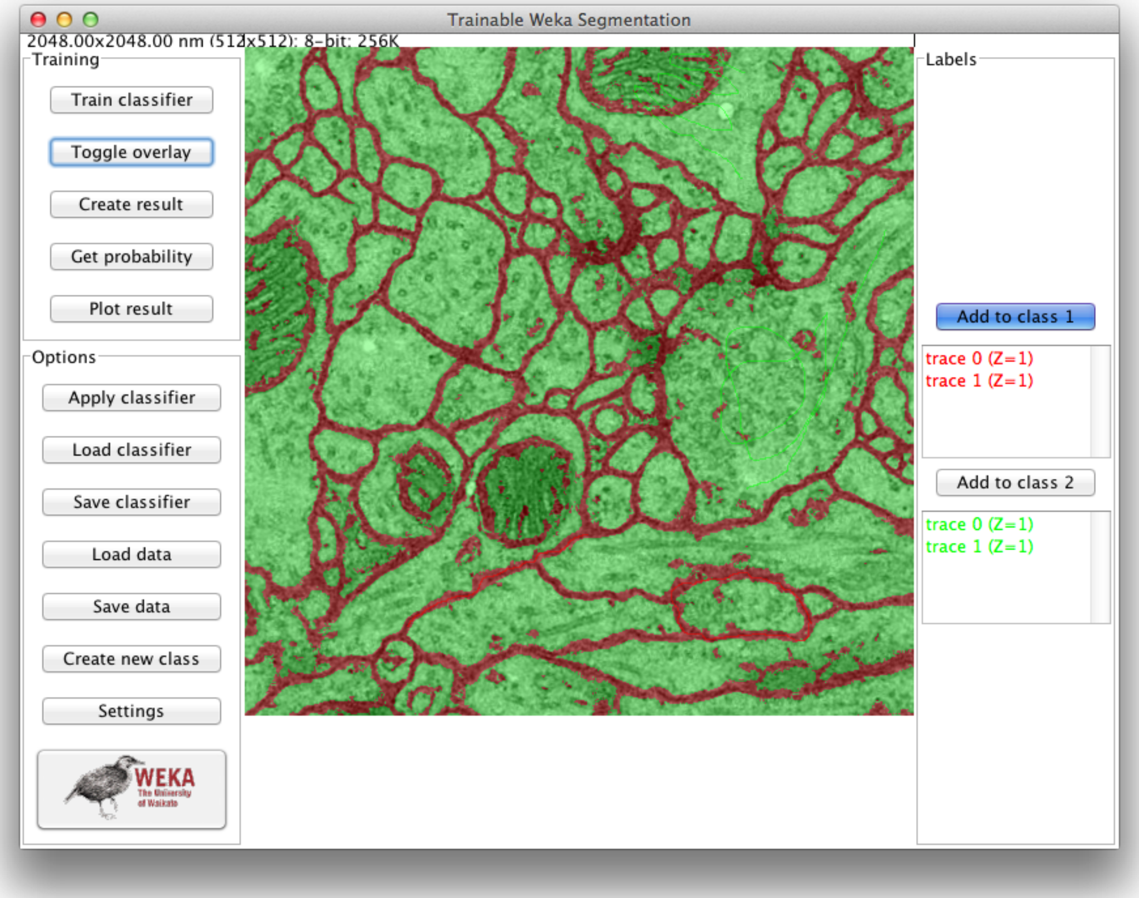


Figure 2: Image segmented with the TWS plugin.

Registration transforms images of different samples or different views of the same sample into a common coordinate system, i.e. overlaying them for easier comparison. This is an important operation in biological image analysis, for example when tiled grids of images are acquired across a sample, but the complete sample is to be analyzed. The Stitching plugin (Preibisch 2009) is a popular ImageJ plugin for combining such image collections into a single cohesive output (see figure 3). The Register Virtual Stack Slices plugin (http://fiji.sc/Register_Virtual_Stack_Slices) provides similar functionality, with the option to transform around a base reference image, and offers a variety of registration techniques (including the elastic implementation from Arganda-Carreras 2008).

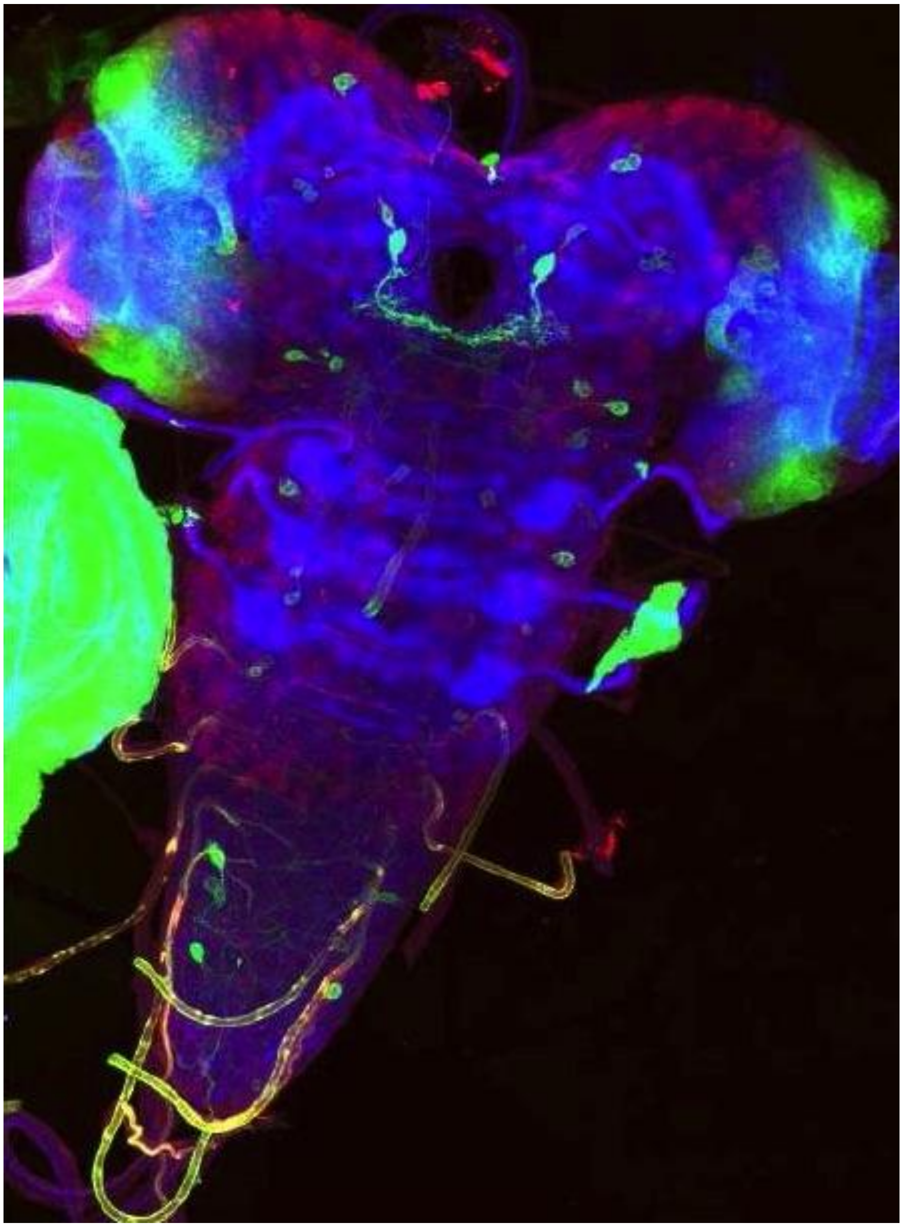


Figure 3: Drosophila larval nervous system, stitched from a 2 x 3 grid of images.

Finally, **tracking** in image analysis follows structures of interest (typically identified via segmentation) over time. While some of these techniques rely heavily on manual interaction, others can be made automatic and robust enough for high-throughput analysis. The ToAST plugin (Hegge 2009) is unique in that it was derived from another tracking plugin, MTrack2 (http://imagej.net/MTrack2), specifically for malaria research but was generalized for binary images. TrackMate (http://imagej.net/TrackMate) is one of the most extensive plugins for ImageJ, providing a robust tracking interface for users, and a consumable API for developers to extend TrackMate itself.

Image acquisition. Before images can be analyzed using the techniques above, they must be produced somehow. While ImageJ has proven to be a valuable tool for scientific fields as diverse as paleontology or astronomy, it is especially popular in biomedical image processing. After all, it was originally developed with images obtained using light microscopes in mind, testament to which is the ImageJ icon, a beautiful Hartnack microscope. It is not surprising, then, that already very early in ImageJ's history scientists wished to use ImageJ not only to process and analyze images, but also to acquire them. Starting with special-purpose plugins supporting certain Hamamatsu cameras or the Scion frame grabbers, scientists developed support to control cameras from within ImageJ and to acquire images. However, with each new plugin came a new and different way to do essentially the same: control microscope hardware. The μManager project (Stuurman 2007) was started to unify the control interfaces, and it became a boon for many a microscopy facility tired of training and retraining scientists in using an ever-increasing number of user interfaces developed by competing hardware vendors. Being open source, μManager enjoys frequent contributions from scientific and hardware vendor developers' alike adding support for more and more hardware such as shutters, stages and cameras. It also proves invaluable as the starting point for new imaging technologies—e.g., the OpenSPIM platform

(Pitrone 2013) to make Selective Plane Illumination Microscopy (SPIM) accessible for everyday use in life science and to develop the technology further.

Development paradigms. Although functionality is an obvious criteria for differentiating plugins, it is also important to consider the various motivations for plugin development and how each contributes to the greater ecosystem. Many plugins (such as Stitching mentioned above) were developed for a specific publication and are offered in the interest of facilitating science. These typically add new functionality to ImageJ, and are freely distributed as a contribution to the scientific community. Depending on how these plugins are funded, they may or may not continue to be supported over time with fixes and optimizations. In contrast, other plugins are actively developed and maintained, often supporting an ongoing scientific project. The goal of such tools is typically broader, leading to the creation of tool suites capable of solving a wide variety of tasks. As a result, these plugin suites—e.g., those of LOCI, MOSAIC and BioVoxxel—require a significant commitment of resources to develop and maintain: often one or more full-time developer, backed by a university or commercial research group.

Underlying all of these plugin paradigms is the concept of openness. As ImageJ itself is provided freely for the benefit of the scientific community, many projects open their source code and welcome community input and feedback. However, there is also a large range of plugins that are available free of cost, but without source code; often accompanied by a license that restricts or prohibits modification and redistribution. Such plugins are often met with wariness in the community—as closed code creates a “black box” with unknown functionality, which thus cannot truly be reproduced. That said, it appears more and more formerly closed projects are recognizing this need and opening up to collaborate and interoperate with other scientific groups and software, for the greater benefit of scientific research (for example, the Biomedical Imaging Group Differentials plugin, and ImageScience plugins both became openly available in September, 2010 thanks to the ImageJ community).

Although ImageJ itself, as a product of the US government, resides in the public domain free of copyright limitations, there is space in the ImageJ ecosystem for commercial interests as well. For example, bridges to commercial software allowing ImageJ plugins in Imaris (http://bitplane.com/imaris/imaris) and MATLAB (http://mathworks.com/products/matlab/) are very much in line with the collaborative spirit of the ImageJ community. Also, independent consultants and companies have emerged, who write special-purpose ImageJ plugins for hire, offer training in general-purpose image processing based on ImageJ, or even training for highly-specialized, advanced image analysis—e.g., using the Trainable Segmentation. There are also hybrid groups mixing scientific and commercial interests, such as OMERO (Allan 2012)—an image database system—and KNIME (Berthold 2008)—a data analytics platform—which offer their software as open source but also have a commercial arm offering paid consulting. The ImageJ community benefits from these sources of users and developers: there are public contributions intended to facilitate contract work (e.g., http://git.io/egvhIw), as well as instances of private work which are subsequently distributed as open source plugins, such as Bob Dougherty’s OptiNav (http://optinav.com/imagej.html).

While the plugins described above positively interact with each other—such as running preprocessing macros to prepare an image for registration—most of them are not explicitly tied to one another. Rather, there are a wealth of available plugins from which to choose, allowing users to select what is most appropriate for each particular task.

Co-operational evolution of software components

At first glance, the relationship between ImageJ and its plugins may appear *commensal*: the ImageJ platform benefits its plugins, but not necessarily vice versa. However, what makes ImageJ a true ecosystem, not just a platform for extension, is the mutual evolution of the application, the plugins and the community. There are several related phenomena which exemplify this sort of mutuality.

The first instance of mutuality is when the needs of a plugin influence the development directions of the core platform. For example, the ObjectJ project (http://simon.bio.uva.nl/objectj/) offers facilities to annotate, segment, track and analyze time-series images. ObjectJ was developed using ImageJ's then-burgeoning macro language, which at the time was not powerful enough to do everything ObjectJ needed. Consequently, ImageJ's macro capabilities matured into a more powerful programming tool partly due to feature requests made by ObjectJ's developer.

Another mutuality scenario is when a useful feature originally developed for a plugin proves generally useful enough to migrate "upstream" into the core application itself. Such occurrences are extremely common within the ImageJ ecosystem. For instance, the TurboReg plugin (http://bigwww.epfl.ch/thevenaz/turboreg/) defined the concept of point selections, which proved so useful they eventually moved to the ImageJ core application, to the benefit of all future plugins. Another example is the Image5D plugin (http://imagej.net/Image5D) for visualization of five-dimensional images—X, Y, Z, time and channels—which became so popular that it inspired ImageJ to introduce analogous "hyperstack" and "composite image" functionality into its core. A third case is the Fiji project (Schindelin 2012), which implemented many components that have since migrated into core ImageJ, including the Command Finder, Script Editor, ImageJ Launcher, and many others.

A third sort of mutuality is when some functionality is so broadly useful that it makes sense to package it as its own software library, independent from ImageJ. In this way, the library's capabilities can be harnessed by ImageJ plugins, but also by software from other ecosystems. Several major software libraries have emerged this way:

Bio-Formats. The Bio-Formats project (Linkert 2010) was started out of the need to open images saved in proprietary image file formats, such as the *.lsm* format written by Zeiss' software, or *.oib* by Olympus'. The question of how to read (and sometimes write) data stored in such formats is not a scientific one, yet is essential to scientific research: data acquired by microscopes needs to be accessible in order to be processed and analyzed. Bio-Formats supports a wide range of proprietary formats, as well as standard file formats not generally associated with life science such as movie files—the infrastructure developed to support microscope vendors' proprietary file formats is flexible enough to easily add support for non-scientific file formats, too. Bio-Formats can be used as an image I/O library by other plugins, but it is not strictly required—the Bio-Formats Importer (see http://imagej.net/Bio-Formats) can open any supported data into a general ImageJ structure, which can then be provided as input to any analytic routine.

ImgLib2. The ImgLib2 library (Pietzsch 2012) came into development as it became clear that the scientific community was in need of a powerful n-dimensional data processing library which could allow computer vision experts to implement their algorithms in a very generic way. Historically, ImageJ supported only a handful of data types (unsigned 8-bit, unsigned 16-bit, floating point 32-bit and 24-bit integer-packed RGB), was limited to at most five predefined dimensions (X, Y, Z, channel, time) and required the complete pixel data to live in computer memory. As a consequence: most plugins supporting more than one data type had to implement the same algorithm multiple times (once per supported data type); were unable to handle new image modalities such as spectral lifetime, multiple angles, etc.; and they simply could not handle images larger than the machine's memory could accommodate. All of these issues have been addressed by ImgLib2, making it is possible to write generic algorithms that will run unmodified, even on data types yet to be invented, on an arbitrary number of dimensions and size. Images can be processed without the need to be loaded completely into memory, and can even be stored in remote databases, or generated on-the-fly, without having to modify or recompile the source code of the processing algorithm.

TrakEM2. One of the largest ImageJ plugins, TrakEM2 (Cardona 2012), was designed to work on very large mosaics of EM images that would be assembled automatically and displayed in a manner similar to Google maps: pre-calculated zoom levels allow fluid navigation of datasets comprising tens of gigabytes of data. Intended for following neurons through large EM recordings, TrakEM2 provides a user interface for interactive segmentation and annotation, with an API for further extension for further extension from the community. At the same time, given its scope and utility, TrakEM2 underscored the need for a new answer to the question: how can this plugin get into the hands of the desired users and developers?

Dispersal of software components

With the exception of code distributed as part of ImageJ itself, the classic model for plugin distribution was to require users to individually download the plugin (often from a website hosted by the plugin's developer or institution) and install it manually to a local directory discoverable by ImageJ. Advertising a new plugin to users was limited to the ImageJ mailing list or word of mouth. This complete scenario was fairly fragile: websites could be taken down, messages buried in archives, and manual plugin installation is just another opportunity for error (especially for plugins with third-party dependencies). Considering these factors, it is not surprising that distributions of carefully curated plugin collections became commonplace.

One of the first plugin collections was known as **MBF ImageJ**. Assembled by Tony Collins at Babraham, later maintained at the Wright Cell Imaging Facility and then at McMaster's Biophotonics Facility, it provided a large number of hand-selected plugins that proved useful for working with light microscopy images. Perhaps more significantly, it also included a coherent manual, making the collection extremely popular—until its website became defunct in 2012.

MBF ImageJ inspired the **Fiji** distribution of ImageJ, including a community-driven documentation wiki at http://fiji.sc/, and later the Fiji Cookbook (http://fiji.sc/Cookbook): a collection of image analysis "recipes" closely modeled after the original MBF ImageJ manual for microscopy. Fiji originally intended to address the need of neuroscientists—later including cell biologists, too—to distribute a number of ImageJ plugins highly relevant to their research that are developed actively, but independently from each other. This required the design and implementation of a general-purpose mechanism to keep all plugins conveniently up-to-date. For technical reasons, it also required the implementation of a general-purpose ImageJ launcher, and led to the development of several other non-neuroscience-specific features such as a Command Finder, and a plugin for submitting bug reports.

Over time, projects such as the 3D Viewer, the Simple Neurite Tracer and bUnwarpJ (Arganda-

Carreras 2008) came under the Fiji distribution umbrella. To accelerate the development of new software components, an extensible scripting framework was designed within Fiji, allowing power users to write and execute scripts written in Python, Ruby, Javascript and other programming languages. Today, Fiji has solidified its role as the gold standard for plugin distribution (see http://fiji.sc/Fiji%27s_Menu) within the ImageJ community. For plugins not directly bundled with Fiji, a robust update site mechanism (see http://imagej.net/Update_Sites) provides an effective venue for developers to publicize their works, from which users can pick and choose new features to install. Since the introduction of the update sites feature two years ago, over 100 public update sites have been created (http://sites.imagej.net/, http://imagej.net/List_of_update_sites).

Mutations of ImageJ

In software development terms, changing an existing program, developing it independently from the original project, is called *forking*—in evolutionary terms, a mutation. It is sometimes frowned upon because it divides forces and sometimes improvements are not shared between forked projects. However, it is important to realize that every time software is modified, every time a local development version differs from the official (or latest-released) version, it is technically a forked project. The challenge is to reconcile the changes at appropriate times—e.g., after completing a new feature or a bug fix. Again, the biological analogy is apt: projects of the same "species" have relatively minor differences and can share improvements or merge back together later—i.e., "interbreed" to produce an offspring with the strengths of both—whereas projects with major differences may ultimately diverge into different incompatible and competing species.

Just as with speciation, software project forks occupy a spectrum of kinship. On one end are projects like ImageJA (http://imagej.net/ImageJA), a very “shallow” fork of ImageJ that is kept closely synchronized with the original, and exists only to provide an unified revision history and build system. Years ago, ImageJA diverged further from ImageJ, providing several technical improvements such as improved support for applets, but most of those changes have since migrated into ImageJ itself, resulting in ImageJA and ImageJ essentially becoming the same animal again.

On the other end of the spectrum are “diverging” forks such as Bio7 (Austenfeld 2012)—an integrated development environment for ecological modelling, scientific image analysis and statistical analysis—and SalsaJ (http://euhou.net/index.php/salsaj-software-mainmenu-9)—software that facilitates teaching astronomy in the classroom. Diverging from ImageJ allows these programs to develop a distinct user experience: e.g., SalsaJ hides image processing methods unsuited for astronomy images to make typical operations more accessible, while Bio7 enhances ImageJ’s programming capabilities by embedding the ImageJ user interface within a rich client platform. The downside is that bug-fixes and overall improvements to ImageJ’s code base do not trickle down to such forks automatically, and require a substantial maintenance effort by the fork’s developers. Even when the software is open source—as is the case with Bio7—there is significant work required to transfer or “port” changes between the two projects, which can elicit increasingly infrequent updates and eventual speciation of the project.

ImageJ2: a macromutation

Over the last fifteen years, ImageJ has grown organically as requested features have been added, with many contributions from outside developers. The result has been a program with a wide range of functionality capable of solving a diverse collection of image processing and analysis problems, particularly in the life sciences. However, this pattern of growth, even when carefully managed, is no substitute for a holistically engineered package built from the ground up with a modular design. Any successful software project, after a period of sustained growth and the addition of functionality outside the scope of the program's original intent, benefits from an extended period of examination and refactoring, and ImageJ is no exception.

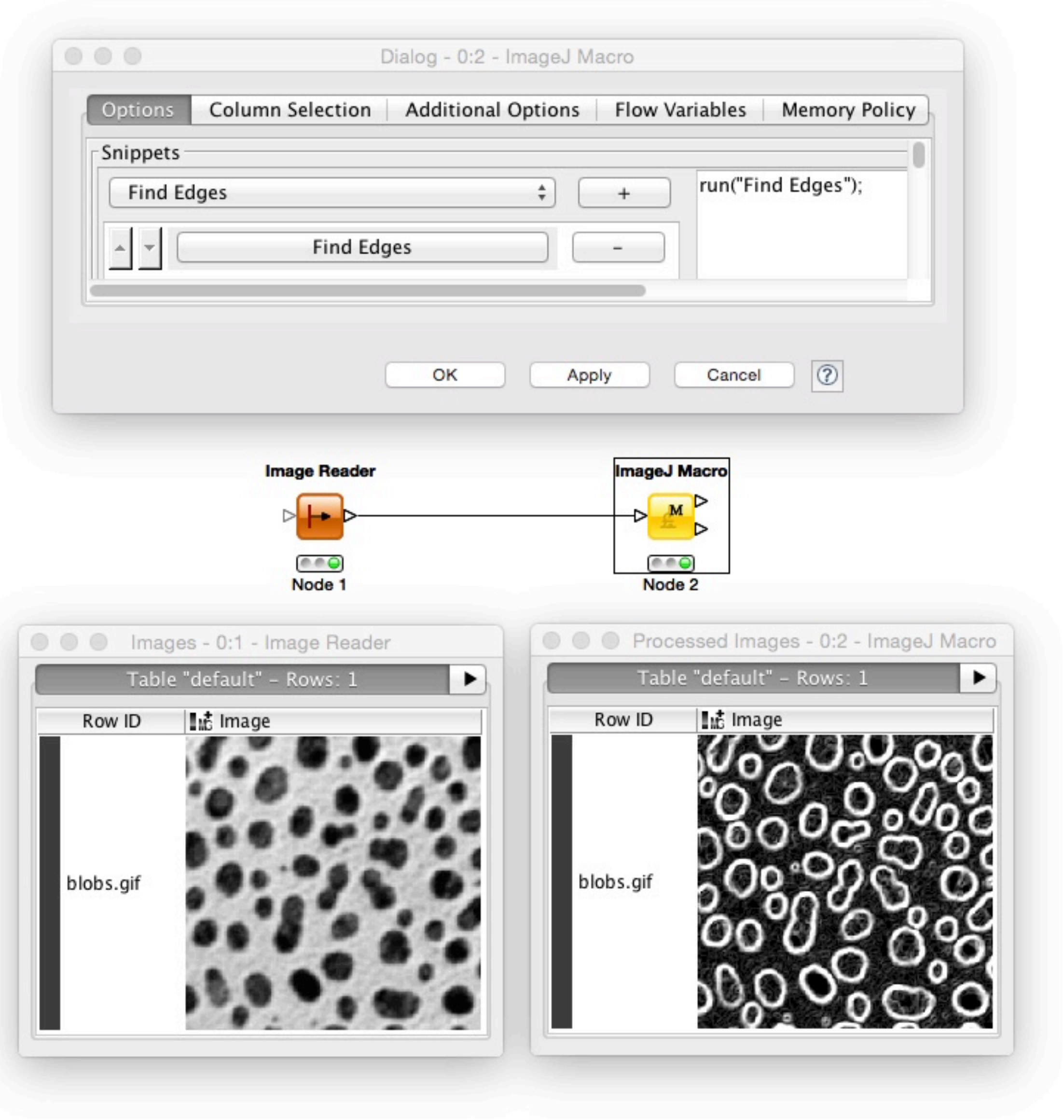
In 2009, NIH funded ImageJ2 (Rueden, *in preparation*), a project to create a new version of ImageJ better able to handle the next generation of multidimensional image data. ImageJ2 provides a unified way to call ImageJ commands from other (even non-Java) software applications, supporting unlimited data types and sizes driven by the ImgLib2 library, as well as extensible data I/O driven by the SCIFIO library (http://scif.io/), a generalization of Bio-Formats beyond the life sciences. In evolutionary terms, ImageJ2 can be seen as a *macromutation* or *adaptation* of ImageJ: a large jump with many internal changes and improvements, intended to better serve the needs of the community. In particular, ImageJ2 provides an extensible data engine intended to accommodate new imaging paradigms such as combined spectral lifetime imaging (SLIM), selective plane illumination microscopy (SPIM), polarized light microscopy, and large tiled image mosaics. The project is backed by a robust software design intended to "future-proof" ImageJ as technology continues to advance. The challenge has been to "harden" the ImageJ application without alienating or fragmenting its widespread user base that relies on it as an everyday research tool. As such, much of the development of ImageJ2 has stressed not only innovation but also compatibility with legacy code, placing a strong emphasis on supporting existing macros and plugins, so as to avoid disrupting established uses of ImageJ as a scientific tool, while at the same time offering a wholly redesigned set of software libraries to meet the demands and challenges of current and future research.

ImageJ2 also addresses the concern of continuity: ImageJ was originally maintained and developed by a single person, now enjoying retirement. Exploring new software development and project management techniques such as distributed source code control, standardized dependency management, automated regression testing, continuous integration and public issue tracking systems, ImageJ now meets the demands of scalable and distributed software development, facilitating its continued development and maintenance.

SciJava: mutualistic symbiosis

As is typical in the realm of computer software, there are many applications with similar, overlapping or complementary functionality to ImageJ, but designed with distinct and diverging requirements. These applications exist at the periphery of the ImageJ ecosystem, often centered within ecosystems of their own. They range from very general-purpose toolkits—such as R (http://r-project.org/), VisAD (Hibbard 1998) and MATLAB—to more domain-specific packages—such as Icy (de Chaumont 2012), Vaa3D (Long 2012) and Endrov (Hendriksson 2013) in the field of bioimage informatics. Many of these tools have a focus different than image processing, but are still part of the ImageJ ecosystem by providing and consuming functionality from each other. As such, users often combine multiple such programs to achieve results not possible with a single software package alone (see figure 4). This approach is particularly effective when the respective software tools are built with this sort of interoperability in mind:

e.g., some such projects—including CellProfiler (Carpenter 2006), MiToBo (Möller 2011), KNIME and OMERO—already support execution of ImageJ commands within their own paradigms, which we foresee applications increasingly doing in the future. Even some purely commercial software packages—such as Imaris and MATLAB via MIJ (Sage 2012)—have facilities for invoking ImageJ and its plugins.



```
run("Find Edges");
```

Table "default" - Rows: 1

Row ID	Image
blobs.gif	

Table "default" - Rows: 1

Row ID	Image
blobs.gif	

Figure 4: Executing ImageJ macro code in a KNIME workflow

One effective tactic for fostering this sort of valuable interoperability is the “hackathon”: an extended brainstorm session bringing together programmers of related scientific projects, with the goal of uninterrupted, high energy exchange of ideas and source code. In a sense, a hackathon is a highly condensed form of how ImageJ’s community operates. During one such hackathon in 2011, the SciJava project (http://scijava.org/) was born: a pledge across several different software projects to work together, share common code, and foster interoperability with one another. One major outcome of this pledge has been that non-image-specific functionality which was originally developed as part of ImageJ2 and Fiji has now been generalized into components of SciJava. Of particular note: ImageJ2’s application container, improved plugin mechanism, and parameterized module framework became the basis for the SciJava Common library; and Fiji’s support for various scripting languages became the SciJava scripting framework. Both ImageJ2 and SCIFIO are now built on the SciJava Common library, meaning they share the same foundation, improvements to which benefit both projects. And some ImageJ plugins have begun to take advantage of the SciJava plugin framework as well: for example, TrackMate uses it so that its segmentation, linking and tracking steps are configurable via plugins.

The future of the ImageJ ecosystem

With the advent of the next generation ImageJ, a number of pressing issues have been addressed, including a key question: “Will legacy macros and plugins still work?” The answer is yes, because ImageJ must support the most important part of the ImageJ ecosystem: its thriving community—including the vast number of legacy macros and plugins it made available. We foresee a steady shift to ImageJ2—e.g., by using the new plugin architecture which not only makes writing plugins easier than before, but also allows the same plugins to be called from other software applications. We also foresee more and more software applications adding a layer of interoperability with ImageJ, particularly its ImageJ OPS library (http://imagej.net/OPS) which provides unified interfaces for basic image manipulations—i.e., a *lingua franca* for implementations of image processing algorithms. Such interoperability has been made substantially easier by the SciJava project, which offers a generic, easily adaptable infrastructure for supporting plugins, among many other convenience functions. We expect the SciJava project to play a more fundamental role in scientific software development beyond ImageJ. It is already in use by core ImageJ2 components, including SCIFIO and ImageJ OPS, to provide powerful extension points. We envisage that the same mechanism could be used e.g. to extend the Trainable Segmentation plugin to support user-provided features. Driven by the demand of concrete scientific software development, we expect SciJava to be developed and used in useful and surprising ways, to provide functionality required in a wide range of software projects.

Although many scientific image processing software projects already are able to interact with ImageJ in one form or another, interoperability is still an ongoing struggle. ImageJ has started to be able to use the power of advanced image processing libraries such as ITK (Benmansour [2012], Yoo 2002), VTK (Sacha 2003, http://ij-plugins.sf.net/), VIGRA (Köthe 2000) and OpenCV (Bradski 2000). But further work is still needed, and the opposite direction—calling ImageJ from said libraries—still has to be developed as well. There are many more active scientific software projects which cannot yet interoperate. Rather than “reinventing the wheel,” it must be the emphasis in the next years of scientific software development to combine efforts and make it possible to benefit from one another, for the common goal: to support scientific research.

References

Allan C, Burel J, Moore J, Blackburn C, Linkert M, Loynton S, MacDonald D, Moore WJ, Neves C, Patterson A, Porter M, Tarkowska A, Loranger B, Avondo J, Lagerstedt I, Lianas L, Leo S, Hands K, Hay RT, Patwardhan A, Best C, Kleywegt GJ, Zanetti G, Swedlow JR: **OMERO: flexible, model-driven data management for experimental biology.** *Nature Methods* 2012, **9(3):**245-253. http://scholar.google.com/scholar?cluster=5929293844302699532&hl=en&as_sdt=0,50

Arganda-Carreras, I. (2008). **bUnwarpj: Consistent and elastic registration in imagej, methods and applications.** In *Second ImageJ User & Developer Conference*. http://scholar.google.com/scholar?cluster=15310324253540786426&hl=en&as_sdt=0,50

Austenfeld, M., & Beyschlag, W. (2012). **A Graphical User Interface for R in a Rich Client Platform for Ecological Modeling.** *Journal of Statistical Software*, *49*(4), 1-19. http://scholar.google.com/scholar?cluster=13595381325241049266&hl=en&as_sdt=0,50

Benmansour, F., Longair, M., Turetken, E., Schindelin, J., and Fua, P. (2012). **ITK Wrapper for Fiji.** Bioimage Analysis Workshop, 2012, Barcelona, Spain. http://bigwww.epfl.ch/eurobioimaging/abstracts/poster-28.pdf

Berthold MR, Cebron N, Dill F, Gabriel TR, Kötter T, Meinl T, Ohl P, Sieb C, Thiel K, Wiswedel B: **KNIME: The Konstanz information miner.** *Data Analysis, Machine Learning and Applications* 2008, 319-326. http://scholar.google.com/scholar?cluster=16514357352138177203&hl=en&as_sdt=0,50

Bradski, G. (2000). **The OpenCV library.** *Doctor Dobbs Journal*, *25*(11), 120-126. http://scholar.google.com/scholar?cluster=5626157233770749l7&as_sdt=5,50&sciodt=0,50&hl=en

Burger, W., & Burge, M. J. (2010). **Principles of Digital Image Processing**. *Springer*. http://scholar.google.com/scholar?cluster=6017806868802164994&hl=en&as_sdt=0,50

Cardona, A., Saalfeld, S., Schindelin, J., Arganda-Carreras, I., Preibisch, S., Longair, M., Tomancak, P., Hartenstein, V. and Douglas, R. J. (2012). **TrakEM2 software for neural circuit reconstruction.** *PLoS One*, *7*(6), e38011. http://scholar.google.com/scholar?cluster=2577882229249708240&hl=en&as_sdt=0,50

Carpenter AE, Jones TR, Lamprecht MR, Clarke C, Kang IH, Friman O, Guertin DA, Chang JH, Lindquist RA, Moffat J, Golland P, Sabatini DM: **CellProfiler: image analysis software for**

identifying and quantifying cell phenotypes. *Genome Biology* 2006, **7:**R100.
http://scholar.google.com/scholar?cluster=11680963210515374635&hl=en&as_sdt=0,50

de Chaumont F, Dallongeville S, Chenouard N, Hervé N, Pop S, Provoost T, Meas-Yedid V, Pankajakshan P, Lecomte T, Montagner YL, Lagache T, Dufour A, Olivo-Marin J: **Icy: an open bioimage informatics platform for extended reproducible research.** *Nature Methods* 2012, **9(7):**690-696.
http://scholar.google.com/scholar?cluster=8170444028478407032&hl=en&as_sdt=0,50

Collins, T. J.: **ImageJ for microscopy.** BioTechniques, Vol. 43, No. S1, July 2007, pp. S25–S30
http://scholar.google.com/scholar?cluster=10897588441048595547&hl=en&as_sdt=0,50

Cristy, J., Bergougnoux, K., Bogart, R., Peterson, J. W., Brown, N., Chiarappa, M., Crimmins, T.R. & Zimmermann, A. (1994). **ImageMagick**. *Currently available at http://ftp.x.org/contrib/applications/ImageMagick/.*
http://scholar.google.com/scholar?cluster=9300835589200007397&hl=en&as_sdt=0,50

Delgado-Gonzalo, R., Thevenaz, P., Seelamantula, C. S., & Unser, M. (2012). **Snakes with an ellipse-reproducing property.** *Image Processing, IEEE Transactions on*, *21*(3), 1258-1271.
http://scholar.google.com/scholar?cluster=12676209567877877204&hl=en&as_sdt=0,50

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., Tomancak, P. & Carpenter, A. E. (2012). **Biological imaging software tools.** *Nature methods*, *9*(7), 697-710.
http://scholar.google.com/scholar?cluster=2521854808864796183&hl=en&as_sdt=0,50

Forster, B., Van De Ville, D., Berent, J., Sage, D., & Unser, M. (2004). **Complex wavelets for extended depth-of-field: A new method for the fusion of multichannel microscopy images.** *Microscopy Research and technique*, *65*(1-2), 33-42.
http://scholar.google.com/scholar?cluster=6212747914090618750&hl=en&as_sdt=0,50

Henriksson, J., Hench, J., Tong, Y. G., Johansson, A., Johansson, D., & Bürglin, T. R. (2013). **Endrov: an integrated platform for image analysis.** *Nature methods*, *10*(6), 454-456.
http://scholar.google.com/scholar?cluster=16199569576064702120&hl=en&as_sdt=0,50

Hegge S, Kudryashev M, Smith A, Frischknecht F. (2009). **Automated classification of *Plasmodium* sporozoite movement patterns reveals a shift towards productive motility during salivary gland infection.** *Biotechnology Journal*, 4, 903-913.
http://scholar.google.com/scholar?cluster=1893935470646292621&hl=en&as_sdt=0,50

Hibbard, B. (1998). **VisAD: Connecting people to computations and people to people.** *ACM SIGGRAPH Computer Graphics*, *32*(3), 10-12.
http://scholar.google.com/scholar?cluster=8209872667707311604&hl=en&as_sdt=0,50

Kaynig, V., Fuchs, T., & Buhmann, J. M. (2010, June). **Neuron geometry extraction by perceptual grouping in sstem images.** In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on* (pp. 2902-2909). IEEE. http://scholar.google.com/scholar?cluster=5799076437717317441&hl=en&as_sdt=0,50

Konstantinides, K., & Rasure, J. R. (1994). **The Khoros software development environment for image and signal processing**. *Image Processing, IEEE Transactions on*, *3*(3), 243-252. http://scholar.google.com/scholar?cluster=14611467212034681378&hl=en&as_sdt=0,50

Köthe, U. (2000). *Generische Programmierung für die Bildverarbeitung*. Dissertation, Fachbereich Informatik, Universität Hamburg, ISBN: 3-8311-0239-2. http://hci.iwr.uni-heidelberg.de/Staff/ukoethe/papers/#PhD

Linkert, M., Rueden, C. T., Allan, C., Burel, J. M., Moore, W., Patterson, A., ... & Swedlow, J. R. (2010). **Metadata matters: access to image data in the real world.** *The Journal of cell biology*, *189*(5), 777-782. http://scholar.google.com/scholar?cluster=17774693189715989612&hl=en&as_sdt=0,50

Long, F., Zhou, J., & Peng, H. (2012). **Visualization and analysis of 3D microscopic images.** *PLoS Computational Biology*, *8*(6), e1002519. http://scholar.google.com/scholar?cluster=11746998690283756601&hl=en&as_sdt=0,50

Longair MH, Baker DA, Armstrong JD. (2011). **Simple Neurite Tracer: Open Source software for reconstruction, visualization and analysis of neuronal processes.** *Bioinformatics* 27 (17): 2453-2454.http://scholar.google.com/scholar?cluster=14433745512074152321&hl=en&as_sdt=0,50

Meijering, E., Smal, I., & Danuser, G. (2006). **Tracking in molecular bioimaging**. Signal Processing Magazine, IEEE, 23(3), 46-53. http://scholar.google.com/scholar?cluster=7130412371448449509&hl=en&as_sdt=0,50

Möller, B., Greß, O., & Posch, S. (2011). **Knowing what happened-automatic documentation of image analysis processes.** In *Computer Vision Systems* (pp. 1-10). Springer Berlin Heidelberg. http://scholar.google.com/scholar?cluster=4511422809874540232&hl=en&as_sdt=0,50

Pham, D. L., Xu, C., & Prince, J. L. (2000). **Current methods in medical image segmentation.** 1. *Annual review of biomedical engineering*, *2*(1), 315-337. http://scholar.google.com/scholar?cluster=18005459743008128630&hl=en&as_sdt=0,50

Preibisch, S., Saalfeld, S., & Tomancak, P. (2009). **Globally optimal stitching of tiled 3D microscopic image acquisitions.** *Bioinformatics*, *25*(11), 1463-1465. http://scholar.google.com/scholar?cluster=2608472115102010l2&hl=en&as_sdt=0,50

Pietzsch, T., Preibisch, S., Tomančák, P., & Saalfeld, S. (2012). **ImgLib2—generic image processing in Java.** *Bioinformatics*, *28*(22), 3009-3011.
http://scholar.google.com/scholar?cluster=17114668151840635304&hl=en&as_sdt=0,50

Pitrone, P., Schindelin, J., Stuyvenberg, L., Preibisch, S., Weber, M., Eliceiri, K. W., ... & Tomancak, P. (2013). **OpenSPIM-an open access platform for light sheet microscopy.** *arXiv preprint arXiv:1302.1987*.
http://scholar.google.com/scholar?cluster=9583955231390830998&hl=en&as_sdt=0,50

Sacha, J. P., Cockman, M. D., Dufresne, T. E., & Trokhan, D. (2003, May). **Quantification of regional fat volume in rat MRI.** In *Medical Imaging 2003* (pp. 289-297). International Society for Optics and Photonics.
http://scholar.google.com/scholar?cluster=5419588985289634268&hl=en&as_sdt=0,50

Sage, D., Prodanov, D., Tinevez, J-Y., and Schindelin, J., **MIJ: Making Interoperability Between ImageJ and Matlab Possible.** ImageJ User & Developer Conference, 24-26 October 2012, Luxembourg.
http://imagejconf.tudor.lu/program/poster/makinginter

Schindelin J, Arganda-Carreras I, Frise E, Kaynig V, Longair M, Pietzsch T, Preibisch S, Rueden C, Saalfeld S, Schmid B, Tinevez J, White DJ, Hartenstein V, Eliceiri K, Tomancak P, Cardona A: **Fiji: an open-source platform for biological-image analysis.** *Nature Methods* 2012, **9(7):**676-682.
http://scholar.google.com/scholar?cluster=17249863664147333646&hl=en&as_sdt=0,50

Schmid, B., Schindelin, J., Cardona, A., Longair, M., & Heisenberg, M. (2010). **A high-level 3D visualization API for Java and ImageJ.** *BMC bioinformatics*, *11*(1), 274.
http://scholar.google.com/scholar?cluster=5464881898236181205&hl=en&as_sdt=0,50

Schneider CA, Rasband WS, Eliceiri KW: **NIH Image to ImageJ: 25 years of image analysis.** *Nature Methods* 2012, **9(7):**671-675.
http://scholar.google.com/scholar?cluster=8490320759898335020&hl=en&as_sdt=0,50

Stuurman, N., Amdodaj, N., & Vale, R. (2007). **Micro-Manager: Open Source software for light microscope imaging.** *Microscopy Today*, *15*(3), 42-43.
http://scholar.google.com/scholar?cluster=7938283370027798220&as_sdt=5,50&sciodt=0,50&hl=en

Yoo, T. S., Ackerman, M. J., Lorensen, W. E., Schroeder, W., Chalana, V., Aylward, S., ... & Whitaker, R. (2002). **Engineering and algorithm design for an image processing api: a technical report on itk-the insight toolkit.** *Studies in health technology and informatics*, 586-592.

http://scholar.google.com/scholar?cluster=10830740755452103531&hl=en&as_sdt=0,50

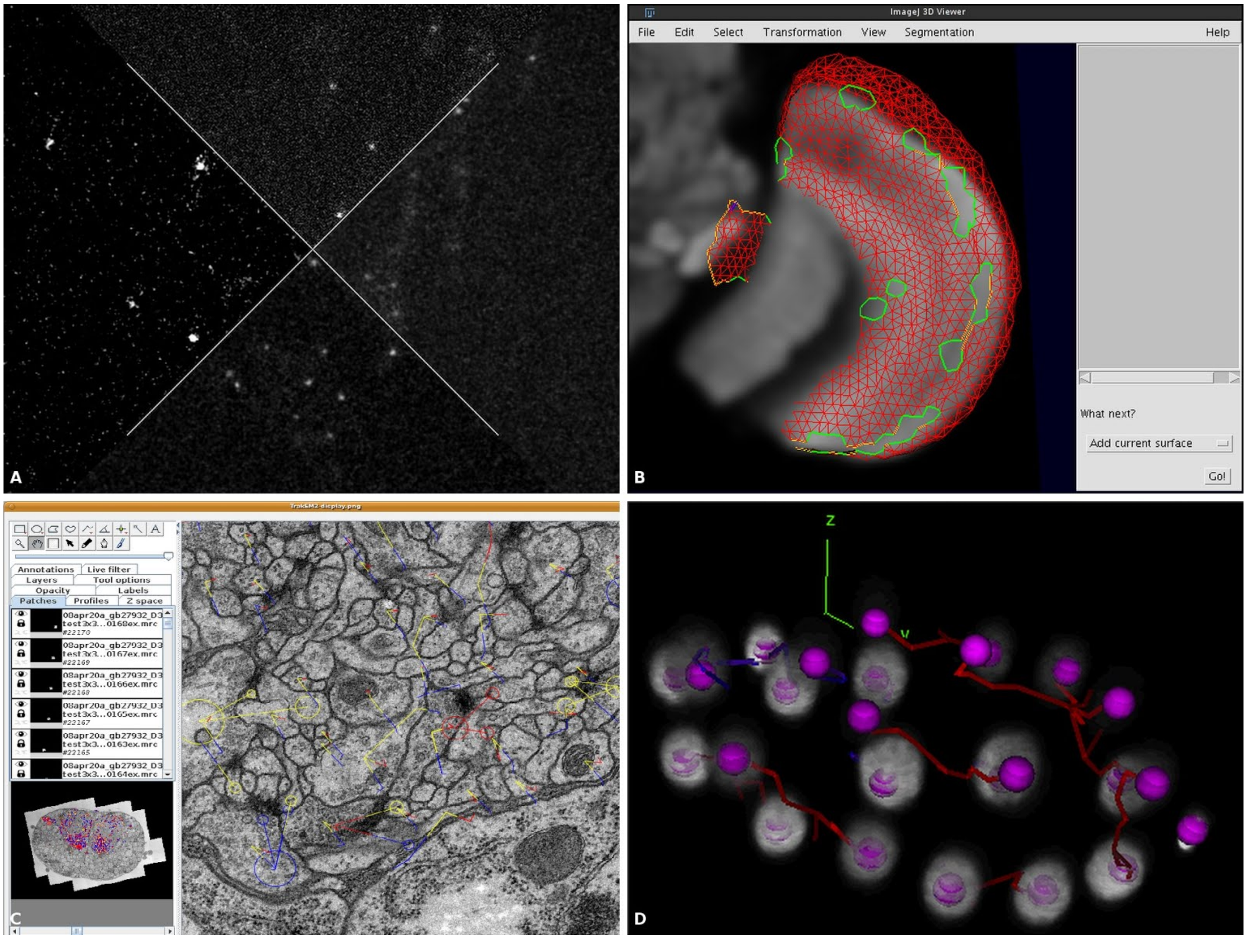

A
ImageJ 3D Viewer
File
Edit
Select
Transformation
View
Segmentation
Help
What next?
Add current surface
Go!
B
TrakEM2 display.png
Annotations
Live filter
Layers
Tool options
Opacity
Labels
Patches
Profiles
Z space
08apr20a_gb27932_D3 test3x3...0168ex.mrc #22170
08apr20a_gb27932_D3 test3x3...0167ex.mrc #22169
08apr20a_gb27932_D3 test3x3...0166ex.mrc #22168
08apr20a_gb27932_D3 test3x3...0165ex.mrc #22167
08apr20a_gb27932_D3 test3x3...0163ex.mrc #22165
08apr20a_gb27932_D3 test3x3...0164ex.mrc
C
Z
Y
D

Figure 1: **A.** example preprocessing stages of imaged boutons (core ImageJ functionality); top: original, right: after denoising, bottom: after background subtraction, left: after contrast enhancement. **B.** active surface segmentation of an adult Drosophila optical lobus (ImageJ 3D Viewer). **C.** registration and annotation of neurons in a third instar larval Drosophila brain (using TrakEM2). **D.** 3D tracking over time of nuclei in C.elegans (TrackMate).

Chapter 3: Analysis of Collagen Fibers in Cancer

Adapted from:
Jeremy S. Bredfeldt, Yuming Liu, Matthew W. Conklin, Joseph M. Szulczewski, David R. Inman, Patricia J. Keely, Robert D. Nowak, Thomas R. Mackie and Kevin W. Eliceiri. Computational Segmentation of Collagen Fibers from Second Harmonic Generation Images of Breast Cancer. Journal of Biomedical Optics.

Second harmonic generation (SHG) imaging can help reveal interactions between collagen fibers and cancer cells. Quantitative analysis of SHG images of collagen fibers is challenged by the heterogeneity of collagen structures and low signal to noise ratio often found while imaging collagen in tissue. Understanding the role of collagen in breast cancer progression can be assessed post acquisition via enhanced computation. To facilitate this, we have implemented and evaluated four algorithms for extracting fiber information such as number, length and curvature from a variety of SHG images of collagen in breast tissue. The image processing algorithms included a Gaussian filter, SpiralTV filter, Tubeness filter, and Curvelet denoising filter. Fibers are then extracted using an automated tracking algorithm called FIRE. We evaluated the algorithms by comparing length, angle and position of the automatically extracted fibers with those of the manually extracted fibers in twenty-five SHG images of breast cancer. We found that the curvelet denoising filter followed by FIRE, a process we call CT-FIRE, outperforms the other algorithms under investigation. CT-FIRE was then successfully applied to track collagen fiber shape changes over time in an *in-vivo* mouse model of breast cancer.

Keywords: Breast Cancer, collagen, second harmonic generation, segmentation, image processing, fiber tracking, microscopy

Introduction

1 The extracellular collagen matrix has been found to promote the progression of many types of cancer. However, the underlying reason for the contribution of stromal collagen to cancer progression is not fully understood, and is the subject of intense biomedical research. Much of this research has benefited from the capabilities of laser scanning microscopy techniques, in particular second harmonic generation (SHG) imaging (1), to capture high-resolution, high-contrast images of individual collagen fibers in tissue and *in-vitro* tissue models (2-6).

For example, Conklin et. al. (7) showed that patterns in SHG images of collagen can predict breast cancer patient outcome. Raub et. al. (8) showed that SHG image characteristics can be used to predict bulk mechanical properties of collagen hydrogels, a common *in-vitro* tissue model for studying cancer cell motility. Nadiarnykh et. al. (9) and Watson et. al. (10) found that SHG image characteristics in ovarian tissue provide quantitative discrimination between tumor and benign tissue. Although SHG has been used successfully in these and many other studies, quantification of collagen fiber shape changes remains a difficult challenge, in part due to large heterogeneities in the patterns observed in SHG images of tissue. For example, in the breast tissue images shown in **Figure Captions**

Figure 1, collagen fibers can be described as wavy or straight (Fig. 1A and B), high or low density (Fig. 1C and D), bundled or individually present (Fig. 1E and F). These descriptions are consistent with previously published observations of collagen structures in tissues (7, 11, 12). In addition, depending on imaging parameters such as depth within the tissue, images can have low signal to noise (SNR) and potentially low dynamic range (Fig. 1G, H, I, and J). Quantitative analysis techniques for SHG images of collagen need to provide robust and informative features within this heterogeneous collection of patterns and image qualities. Also, in order to elucidate interactions between cells and individual collagen fibers, effective quantitative analysis techniques should be able to extract information about individual fibers, such as fiber number, length, angle and curvature. The work reported here is motivated by these two requirements, the need for robust performance and the need for fiber-level information in SHG image analysis of collagen.

While these two requirements may be met by manual analysis (13), inter-observer and intra-observer variance can be significant and time requirements prohibitive. Computer assisted image feature extraction is poised to help meet these requirements in an automated fashion. Transform or filter based methods have been used for SHG collagen analysis such as the Fourier transform method published by Falzon (14), the combined Fourier and Hough transform method by Bayan (15), the Curvelet transform method by Pehlke (16), the Fourier and fractal based method reported by Frisch (17), the directional gradient technique suggested by Altendorf (18) , and the grey level co-occurrence method published by Hu (19). These techniques can be highly robust, often able to detect important features in a diversity of image settings. However, since these techniques do not extract individual fibers, they lack the ability to extract fiber-level information. For example, transform based methods can provide general information about fiber size and direction at each point in an image, but cannot determine the actual fiber number, nor the length or curvature of the fibers. Since pixels are not grouped into individual fibers, these methods may not sense the difference between many short, randomly-oriented, straight fibers and long curvy fibers, features that may help to classify patients into high and low risk groups for ovarian (9) and other cancers. In addition, angle distributions generated by these algorithms would generally produce bias toward longer and potentially thicker or brighter fibers, since more pixels are present in longer and thicker fibers and distributions are based on pixels and not fibers. Avoidance of bias such as this is necessary to accurately pursue biological hypotheses that are based on high resolution microscopy image data, for example, the hypothesis that fiber angle distribution may help predict metastatic potential of cancer cells (4, 5, 7, 20).

Fortunately, fiber tracking and extraction methods, such as those published by Wu (21, 22) and Stein(23), have been developed to extract fiber-level information from images of *in-vitro* collagen matrices. These methods can enable the automated measurement of important fiber-level parameters such as fiber length, number and curvature and have been used to estimate collagen gel mechanical properties based on confocal images of stained gels. However they have not been applied to SHG images of collagen *in-situ*. While these approaches are powerful, they often fail to properly segment fibers in the dense or low signal-to-noise (SNR) situations commonly encountered in SHG images of tissue. Examples of two SHG images are shown in Figures 2A and D with corresponding manual fiber extractions shown in Figures 2B and E. The Fiber Extraction (FIRE) algorithm, developed and made available by Stein (23), produces the overly complex fiber network shown in Figure 2C, an erroneous star pattern in Figure 2F and in both cases fails to identify many of the fibers extracted by the human observer.

Instead of using transform based methods or fiber extraction methods alone, a more strategic approach would be to combine complimentary methods and use transform/filter based methods as a preprocessing step to fiber extraction. This combined approach has potential for robust performance in a wide range of challenging imaging situations commonly seen in cancer imaging while simultaneously allowing for fiber-level information to be extracted from images. In this study, we present an approach for integrating transform/filter based preprocessing techniques with fiber extraction and evaluating the performance of our combined approach with the ultimate goal of improving the fiber extraction accuracy of an algorithm such as FIRE. Our hypothesis is that the application of appropriate pre and post image processing algorithms can significantly improve fiber extraction in SHG images. This will enable more accurate fiber angle distributions, thus allowing for increased sensitivity to detect collagen alignment changes related to cancer progression.

We evaluate four candidate preprocessing techniques including the simple Gaussian filter, the SPIRAL-TV filter (24), the tubeness filter (25), and a curvelet transform based denoising filter (26, 27). Other than the Gaussian filter, these filters were chosen based on their published ability to highlight edge information in images while simultaneously suppressing spatially uniform structures and noise. We have chosen to use the FIRE algorithm based on evidence of its ability to extract fibers from *in-vitro* collagen gel networks and its availability (28), however, other fiber extraction tools may be substituted for the FIRE algorithm. We focus our analysis on 2D SHG images because the effective nonlinear susceptibility declines sharply when fibers are tipped out of the imaging plane (29-31), however our methods may naturally extend to 3D without significant alteration.

2 Materials and Methods

Our experimental approach for the evaluation of quantitative collagen fiber extraction is illustrated in Figure 3. Twenty-five images were annotated by three human observers who traced each fiber within each image to create a surrogate for ground truth. The same images are then filtered with one of four image filters, which are described in the following sections. Fiber tracking and extraction is then performed using the FIRE algorithm. We evaluated the automated fiber extraction accuracy by comparing the length, angle and position of each fiber extracted by the FIRE algorithm with each fiber that was manually extracted to determine if fibers can be considered detected, missed, or falsely detected. F-measure scores are created based on these rates and a single parameter in each image filter was adjusted to optimize the F-measure score for each algorithm. The details of each step in this process are given in the following sections. Following algorithm evaluation, we applied the best performing algorithm to quantify collagen fiber shape changes during tumor progression in an *in-vivo* mouse model of breast cancer.

2.1 Sample Preparation

To evaluate algorithm accuracy, we chose to use images of both human and mouse tissues as they are both routinely used by our group and others for studying stromal interactions during breast cancer development. Human ductal carcinoma in-situ biopsy samples were obtained from 2 completely de-identified patients, paraffin processed, sectioned to 5 microns, hematoxylin and eosin (H&E) stained and coverslipped. In addition, tumor bearing mouse mammary glands from 8 week old MMTV-polyoma middle-T (13) mice were harvested and imaged fresh, intact, and unstained. To demonstrate an live biological imaging application of our collagen fiber analysis tools, *in-vivo* images were captured through a glass coverslip intravital imaging window that was surgically placed immediately superficial to palpable tumors within the mammary gland (32) in live 8 week old MMTV-polyoma middle-T (PyMT) mice. Animals were anesthetized while imaging was performed at 8 weeks and 12 weeks of age. All human tissue images were collected following institutional review board approval at the University of Wisconsin at Madison. Mouse tissue images were obtained using protocols approved by the University of Wisconsin at Madison institutional animal use and care committee.

2.2 SHG Imaging

SHG images were captured with an excitation wavelength of 890 nm, a pulse length of approximately 100 fs, and an emission filter centered at 445 nm with a 20 nm bandwidth (Semrock, Rochester, NY). Excitation light was focused onto the sample using a 10X (N.A. 0.5) objective. Pixel size was approximately 0.75 microns and pixel dwell time was approximately 10 microseconds per pixel. Average laser power at the sample was adjusted to approximately 15 mW for slide imaging and 30 mW for intact and *in-vivo* tissue imaging using a Pockel's cell and a polarizer. Forward SHG (FSHG) was used to image slides and backward SHG (BSHG) was used to image intact and *in-vivo* mouse tissue. Emission light was detected with a 7422-40P (Hamamatsu, Japan) photomultiplier tube in both FSHG and BSHG cases. All SHG images of collagen were captured in regions adjacent to mammary ductal epithelium verified by white light images of H&E in the slides and by cellular autofluorescence from FAD in the intact tissue and *in-vivo* cases (13). Single images were captured of the slides and z-series were captured for the intact tissue and *in-vivo* imaging experiments. For intact tissue imaging, representative images were selected from each z-series for quantitative analysis. For *in-vivo* imaging, Z-stacks at 3 imaging locations were captured at each time point. Three images at depths of approximately 5, 10 and 15 microns below the imaging window were selected for analysis at each location for a total of 9 images per time point.

2.3 FIRE Algorithm

We briefly review the FIRE process here. A more detailed description of the algorithm can be found in reference (23). FIRE is an automated tracking method that can extract the geometric structure of three dimensional collagen images and is capable of generating information about the number, length and curvature of the collagen fibers in an image. The first step of this method is to apply a threshold to form a binary image such that foreground pixels represent potential fibers and background pixels represent pixels where no fiber is present. Next, the distance transform on the binary image is performed to yield the distance from each foreground pixel to the nearest background pixel. Then the maximal ridges of the smoothed image formed by the distance transform are searched to create a list of nucleation points. Branches are formed by extending the fiber from each nucleation point based on fiber trajectory. Short fiber branches are then pruned and closely associated fibers are finally linked based on the fiber length, fiber direction and the distance between adjacent fibers. In the associated FIRE software (28), there are about 20 adjustable parameters initialized with default values. There are usually only a few parameters that need to be adjusted, such as those impacting the binary image generation, the search for nucleation points and fiber linkage. To our knowledge, the FIRE method has only been tested on confocal reflectance and confocal fluorescence images of *in-vitro* collagen gels, but has not been applied to extract collagen fibers from SHG images of tissue.

Each preprocessing technique described in this paper was followed by nearly identical implementations of the FIRE algorithm. The only difference is in the threshold used for creating the initial binary image. This threshold was hand optimized to produce the highest quality fiber extractions across all test cases for each algorithm.

2.4 Preprocessing Algorithms

The four preprocessing algorithms evaluated here are described briefly below. More detailed background information on the advanced filters can be found in their respective references.

2.4.1 Gaussian filter

A simple 2-D Gaussian filter (GF) was used as a baseline for comparison against the other more advanced filters. The standard deviation of the simple GF was optimized to produce fiber extractions that most closely matched the human observers using the iterative approach diagramed in Figure 3.

2.4.2 SPIRAL- TV filter

The SPIRAL-TV (SPTV) algorithm, by Harmany et. al. (24) was developed to accurately extract features from images where Poisson noise dominates, a common occurrence in SHG imaging of collagen in tissue or *in-vitro* collagen gels due to the low signal levels often encountered in such imaging experiments (33). This algorithm has applications in compressed sensing, nuclear medicine tomographic reconstruction and super-resolution reconstruction in astronomy. The algorithm iteratively approximates a solution to the constrained optimization problem given by

$$f^{k+1} = \underset{f \in \mathbb{R}^n}{\arg\min}\, F^k(f) + \tau * pen(f)$$

$$subject\ to\ f \geq 0$$

where f is the approximation to the image of interest, $F^k(f)$ is the negative Poisson log-likelihood function at iteration k, and $pen(f) = \|f\|_{TV}$ is the total variation seminorm penalty scheme (34). The scalar parameter τ was optimized to produce the best match when comparing human and automated fiber extractions. SPTV was shown to perform well at highlighting strong edges in images and smooth noise in low gradient areas (24). The designers of this algorithm have tested it on noisy computed tomography reconstruction data however it has not been heretofore applied to preprocessing for fiber extraction from SHG images.

2.4.3 Tubeness filter

The tubeness filter (TF) is an ImageJ plugin implemented by Longair, Preibisch and Schindelin (35) and is based on the work published by Sato et. al. (25). The algorithm highlights fiber-like structures in images while attenuating homogeneous or noisy regions and has found application in processing images of neurons and blood vessels (25, 36). This filter enhances fiber structures by first applying a 2-D Gaussian filter with the standard deviation optimized to produce the best overall fiber extractions. Next, the Hessian is computed at each point in the image and the eigenvalues, λ_1 and λ_2 for the 2-D case, of the Hessian matrix are found. The resulting pixel value is given by the following rule:

$$\lambda_g = \begin{cases} |\lambda_2|, & \lambda_2 < 0 \\ 0, & otherwise \end{cases}$$

To our knowledge, this filter has not been evaluated for its ability to highlight collagen fibers in SHG images of tissue.

2.4.4 Curvelet filter

We have also implemented a denoising filter based on the 2-D curvelet transform (CT). The CT was developed by Candes and Donoho (37) to overcome the missing ability of the conventional wavelet transform to highlight lines and edges. Our group has recently reported on the successful use of the CT for finding fiber alignment information in SHG images of collagen (16). Here we report on the use of the CT as a preprocessing step to fiber extraction. Briefly, the CT represents images as superpositions of elements that are constant along ridgelines and wavelets in the orthogonal direction. Curvelet lengths and widths vary with scale and obey the rule $width \approx length^2$. Simple curvelet coefficient thresholding has been shown to be an improvement over advanced denoising techniques based on wavelets such as decimated or undecimated wavelet transforms (26). Our denoising implementation uses the frequency wrapping version of the Fast Discrete Curvelet Transform (27) and reconstructs images using the top $x\%$ of the curvelet coefficients from the intermediate scales 4, 5, and 6 out of 7 total scales in our test cases. The parameter x was optimized to produce the best overall results as indicated in the block diagram in Figure 3. Scale selection may vary with different applications; however we chose to remove the finest scale (7th scale) due to the high noise content present at this scale. The coarser scales (scales 1-3) did not represent the size of the fibers in our images and were therefore discarded.

2.5 Algorithm integration and evaluation

As shown in Figure 3, in order to yield optimal fiber evaluation results, each filter was optimized in an iterative manner to find the optimal normalization parameters. The FIRE parameters could have also been iteratively optimized, however we decided to fix the FIRE parameters for each of the preprocessing algorithms, except for the initial threshold that separates fiber pixels from background pixels. This threshold was hand optimized for each image as well as for each algorithm. This was necessary because the image histogram of the result of each algorithm was significantly different. The method for evaluating the fiber segmentation was as follows: three human observers were asked to manually segment all fibers in each of the test images into regions of interest (ROI). The images were annotated using the ImageJ ROI Manager. The ROIs for each of the test cases were saved for each of the 3 observers. These ROIs were then read into MATLAB using the Miji toolbox (38). The fibers were extracted by FIRE for each test case, and each algorithm was then compared with the manually extracted fibers for each test case and each observer. A manually segmented fiber was associated with an automated fiber, and vice versa, if the two had similar average angles, similar positions, and similar lengths. The average angle of a fiber was computed by finding the absolute angle of the line connecting the end points of the fiber. Fiber length was computed as the distance traveled along the fiber. Distance between fibers was computed as the sum of the distance between the nearest neighboring points between the manual and automatically segmented fibers. The number of true positive fibers (TP), false positive (FP), and false negative fibers (FN) were then found by counting the number of associated manual fibers, unassociated automated fibers, and unassociated manual fibers respectively for each test case. Precision (also called positive predictive value) and recall (also called sensitivity or true positive rate) were computed as $Precision = TP/(TP + FP)$ and $Recall = TP/(TP + FN)$, and the harmonic sum of the two was computed as follows

$$Fmeasure = 2 * \frac{Precision * Recall}{Precision + Recall}$$

The $Fmeasure$ result for each of the preprocessing algorithms was averaged over all test cases for a given observer, producing $Fmeasure_n$, where n represents observer number. Then, the $Fmeasure_n$ result was averaged over all observers and the standard deviation between observers was computed.

3 Results

Comparing the four image-processing techniques to each other, as shown in row 1 of Figure 4, reveals that edge-preserving filters such as SPTV, although effective for denoising without loss of edge information, do not lend themselves well to improving the fiber tracking results. On the other hand, the TF and CT create ridges along fiber centers (Fig. 4, row 1), helping to ease the difficulty of threshold selection and helping the fiber tracking algorithm to follow the centers of thick or noisy fibers. Examination of fiber tracking results in Figure 4 row 2, shows many completely erroneous fiber tracks for the unprocessed, GF and SPTV filtered cases (red arrows), whereas the TF and CT filtered results show several properly segmented fibers (green arrows). Each of the images in Figure 4 are 128 by 128 pixel regions cropped out of larger images.

To further test the effect of preprocessing on collagen fiber segmentation, the manual segmentation results were compared to the automated fiber extraction results for two representative test cases, shown in Figure 5. Each row in the figure is a different test case, while each column represents a different method of fiber segmentation. Column 1 shows the original images with no overlaid segmentations. Columns 2 through 6 show the original image with overlays of the manual, GF, SPTV filter, TF, and CT filter segmentations respectively, where FIRE was performed following each of the filter preprocessing steps. Although we had three observers manually segment each of the test cases, the manual segmentations shown in column 2 represent the segmentations of a single observer. Each tile in Figure 5 is a 128 by 128 pixel crop of a larger image. The quality of extraction of individual fibers under the various processing conditions was compared (arrows in Fig. 5A and B) where it was found that improperly segmented fibers as a result of GF or SPTV filtering were accurately segmented using the TF or CT filters.

In order to assess the accuracy of collagen fiber segmentation, the results of each of the fiber extraction preprocessing algorithms were compared against each of the three segmentations performed by the independent observers using a collection of custom scripts written in MATLAB (MathWorks, Natick, MA). If a fiber segmented by the automated process had a similar angle, close proximity, and similar length to a manually segmented fiber, then an association was made between the automated and manual fibers, indicating a true positive. After all fibers were evaluated, all remaining unassociated manual fibers were counted as false negatives (misses) and all remaining unassociated automatic fibers were counted as false positives (false hits). Precision, recall and their harmonic sum (F-measure) were computed and compiled for all test cases and all observers. Overall average F-measure scores for each of the preprocessing algorithms are shown in Figure 6. The average F-measure score for the CT filter was the highest followed by the TF, SPTV, and GF. The error bars indicate the standard deviation between the F-measure scores from each of the 3 observers and show that the scores between observers were very similar, meaning that the CT filter result was the closest match to all 3 observers.

After identifying the CT filter as the top performing preprocessing algorithm, we applied the combined CT-FIRE algorithm to the measurement of collagen fiber shape changes in an *in-vivo* mouse model for breast cancer. The results of this study are shown in Figure 7.

Representative images show clear differences in waviness of fibers between the early (Fig. 7A) and late (Fig. 7B) time points. The colored lines overlaid on the images are the automated fiber segmentations produced by CT-FIRE. These overlays qualitatively illustrate the high fiber segmentation quality that can be expected from the CT-FIRE algorithm. Fiber waviness (W) was quantified for each extracted fiber by dividing the distance along the path of the fiber (d_0) by the distance between the end points of the fiber (d_n).

$$W = \frac{d_0}{d_n}$$

Thus for perfectly straight fibers $W = 1.0$ and more wavy fibers $W > 1.0$. We labeled a fiber as wavy if W was greater than a threshold value of 1.08. Then, to compute the wavy-fraction per image, the number of wavy fibers was divided by the total number of fibers found in each image. The resulting wavy-fraction values were averaged over all images at each time point and plotted in the bar graph shown in Fig. 7C. We observe that the fraction of curvy fibers at the 8 week time point was approximately 0.19+/-0.04 and 0.75+/-0.03 at the 12 week time point. The error terms given here and error bars in Fig. 7C represent one standard deviation around the average wavy-fractions for the 9 images analyzed at each time point and indicate that there was close agreement between all images within a given time point.

4 Discussion

In the present study we compare preprocessing approaches prior to the application of the FIRE fiber extraction algorithm to identify fiber-level collagen characteristics in a series of SHG images of collagen in mammary tissue. Fiber extraction facilitates automated analysis of collagen features such as fiber number, length, and curvature. These features are important to researchers studying the role of the extracellular matrix in cancer progression. Computer assisted interpretation of these fiber-level collagen patterns has the potential to generate more reliable and reproducible results compared to manual or transform/filter-based quantification methods. Furthermore, an algorithm that identifies collagen fiber characteristics in tissue samples may enable large scale studies of tumor associated collagen signatures supporting the manual analysis performed previously (7).

To our knowledge, FIRE has not been applied to SHG images of collagen in tissue. According to our testing, FIRE works well in some situations without any preprocessing or pre-filtering. However, the algorithm fails when collagen fibers are densely packed or image quality is degraded, common occurrences while imaging collagen in tissue. Our work aims to extend FIRE's application sphere to include complicated SHG images in tissue and to quantitatively compare the performance of a selection of preprocessing algorithms. Our results show that both the CT and the TF are very promising and improve the fiber extraction accuracy achieved by the FIRE algorithm in many key situations. In addition, we demonstrate the application of our top performing algorithm to extract *in-vivo* fiber curvature changes during the development of a mouse mammary tumor. Although FIRE is used in our study for fiber extraction, other effective approaches that have been developed for vessel segmentation or neural diffusion mapping such as statistical tracking (39-41) may be effective in SHG image analysis. We believe the CT and TF would generally improve these algorithms as well.

A recent review (42) suggested that there might be advantages of CT being applied in combination with other approaches for image processing such as fiber extraction, as we have now demonstrated is the case. By selecting and thresholding the most representative scales, the CT based method shows the best performance for both denoising the image and enhancing edge information producing a better fiber extraction among the proposed preprocessing algorithms discussed in this paper. Evidence supporting this claim is presented in Figure 6 where the overall F-measure result was the highest for the CT method and notably a threefold improvement over a standard GF when considering all 25 test cases analyzed in this study. In addition, the CT based method simplifies the often-difficult choice of selecting a threshold to binarize the image early in the FIRE process. Image thresholding can be difficult in low SNR and non-stationary images but may be alleviated through the application of more complicated thresholding techniques (43) or via the grey level distance threshold (44). In our case, the inverse CT makes grey level threshold selection simple by placing the background on the negative side of zero and the foreground on the positive side of zero, allowing the threshold to always remain at zero. Although grey level thresholding is simplified, CT denoising adds a threshold selection step. We chose a hard thresholding approach since it had robust performance for all cases we tested, though other soft thresholding or scale-adaptive thresholding techniques may be adopted to finely adjust the CT-reconstruction. In addition, to take full advantage of the multiscale analysis of CT based approaches, an optimal scale combination must be obtained according to the features of the images to take into account different fiber width, length and dynamic intensity changes. The CurveAlign software (16) previously developed in our group may be used to show the curvelet centers and directions of the fiber edges at a specified scale, which may be helpful for choosing the optimal scales and threshold of the curvelet coefficients.

The TF method produces slightly lower overall fiber segmentation accuracy compared to the CT method as shown by the lower optimized F-measure score in Figure 6. However, the optimized recall score of the TF method was higher than the CT method. This indicates the use of the TF method if recall is most critical, in other words if priority is placed on not missing real fibers. In this study, we decided to balance recall and precision equally, therefore a missing fiber and a false alarm fiber were considered equally important (see F-measure calculation). When this is taken into account, although close, the CT method produced higher accuracy segmentations on average compared to the TF method.

The GF and SPTV methods produce similarly inferior segmentations compared to the CT and TF methods. One reason for this is that the GF and SPTV methods lack the ability to normalize the fibers in the images, such that bright or dim fibers, thick or thin fibers do not generate the same signal level in the output image. This lack of image normalization in the GF and SPTV methods cause difficulty in the threshold selection step. In addition, these filters do not enhance the ridges along the centers of the fibers, which is an attractive feature of both the CT and TF methods. The GF is able to attenuate high frequency noise, but does not preserve edges. The SPTV method filters high frequency noise and preserves edge information, however allows plateaus of high signal level to remain in the image, such as those seen at the center of bright, thick fibers. For these reasons, the GF and SPTV methods are not ideal for preprocessing SHG images of collagen prior to fiber tracking.

Following our evaluation and comparison of these image processing algorithms, we applied our top performing algorithm, termed CT-FIRE, to quantify collagen fiber shape changes over time in an in vivo mouse model for breast cancer. We observed a significantly larger fraction of highly curved fibers at the late time point compared to the early time point, indicating a quantifiable collagen matrix reorganization in the vicinity of the developing mammary tumor. Although the mechanisms underlying this observed matrix reorganization are currently unknown, we have demonstrated the power of a tool like CT-FIRE to quantify key aspects of these dynamic processes, enabling further studies of ECM remodeling.

This paper focuses on image processing as a technique for quantifying structural information about collagen fibers in SHG images. However, it should be noted that there are a number of related techniques that use information about the polarization or directionality of the SHG signal to make inferences about collagen fiber orientation or estimates of the non-linear susceptibility tensor (31, 45-50). These techniques have the potential drawback of adding costly components to the imaging system and often requiring multiple images to be captured per imaging frame, limiting their usefulness for applications such as *in-vivo* imaging. Our goal here was to establish a robust technique for quantitative collagen architecture analysis of images captured with standard SHG imaging techniques.

It is worth mentioning also that although the CT and TF preprocessing methods can improve the results of the FIRE algorithm to some degree, they may do little about some intrinsic limitations of FIRE, such as the ability to properly segment crossing or cross-linked fibers, extremely curvy fibers, or fibers with gaps due to the fibers that travel in and out of the focal plane as we observed in our testing. However, with the improvements provided by the combined approach of CT-FIRE, we anticipate being able to more accurately measure collagen fiber angle distributions in a highly automated fashion, thereby leading to better understanding of the interactions between cells and collagen fibers. In order to link collagen architecture to cellular features, SHG imaging and CT-FIRE may be combined with complementary imaging techniques such as multiphoton excited fluorescence imaging (51) and fluorescence lifetime imaging (52) which allow imaging of both extrinsic and intrinsic fluorescence of tumor and stromal cells. In the future, accurate assessment of tumor-stromal interactions will help reveal prognosis or treatment response in diseases such as breast cancer.

5 Conclusion

We demonstrate here an integrated approach for quantitative SHG collagen image analysis and algorithm evaluation. We show the application of curvelet transform denoising as a preprocessing step for FIRE fiber extraction, a process we call CT-FIRE, performs more accurate fiber segmentations, compared to other techniques we investigated in a variety of collagen images of human breast and mouse mammary tissue. We then demonstrate that CT-FIRE can automatically sense changes in collagen fiber curvature from images captured in an *in-vivo* breast cancer mouse model. Our current work uses both MATLAB and Fiji (35) ImageJ image processing tools in combination. To make these approaches more widely accessible, we plan to develop a single Fiji plug-in to perform the CT-FIRE process to produce 2D and 3D collagen fiber network extractions. Other future efforts will include the evaluation of multiple fiber tracking algorithms applied to collagen fiber tracking in SHG images. Although this study focused solely on breast cancer, the use of these fiber quantification techniques should be easily adapted to SHG images of other collagen related diseases. A MATLAB implementation of the CT-FIRE algorithm is available at http://loci.wisc.edu/software/ct-fire.

Acknowledgements

We would like to acknowledge assistance and valuable discussions with members of Keely, Medical Devices and LOCI groups. We also acknowledge funding from NIH R01 grants CA114462 and CA136590 to P.J.K and K.W.E, T32 CA009206 to J.S.B. and the Morgridge Institute for Research.

6 References

1. P. J. Campagnola, M. D. Wei and L. M. Loew, "Second harmonic generation imaging of living cells," *Biophysical Journal* 76(1), A95-A95 (1999)

2. W. R. Zipfel, R. M. Williams, R. Christie, A. Y. Nikitin, B. T. Hyman and W. W. Webb, "Live tissue intrinsic emission microscopy using multiphoton-excited native fluorescence and second harmonic generation.," *Proc Natl Acad Sci U S A* 100(12), 7075-7080 (2003)

3. G. Cox, E. Kable, A. Jones, I. K. Fraser, F. Manconi and M. D. Gorrell, "3-dimensional imaging of collagen using second harmonic generation," *Journal of Structural Biology* 141(1), (2003)

4. J. Condeelis and J. E. Segall, "Intravital imaging of cell movement in tumours.," *Nat Rev Cancer* 3(12), 921-930 (2003)

5. M. Sidani, J. Wyckoff, C. Xue, J. E. Segall and J. Condeelis, "Probing the microenvironment of mammary tumors using multiphoton microscopy.," *J Mammary Gland Biol Neoplasia* 11(2), 151-163 (2006)

6. P. P. Provenzano, K. W. Eliceiri, L. Yan, A. Ada-Nguema, M. W. Conklin, D. R. Inman and P. J. Keely, "Nonlinear optical imaging of cellular processes in breast cancer.," *Microsc Microanal* 14(6), 532-548 (2008)

7. M. W. Conklin, J. C. Eickhoff, K. M. Riching, C. A. Pehlke, K. W. Eliceiri, P. P. Provenzano, A. Friedl and P. J. Keely, "Aligned collagen is a prognostic signature for survival in human breast carcinoma.," *Am J Pathol* 178(3), 1221-1232 (2011)

8. C. B. Raub, V. Suresh, T. Krasieva, J. Lyubovitsky, J. D. Mih, A. J. Putnam, B. J. Tromberg and S. C. George, "Noninvasive assessment of collagen gel microstructure and mechanics using multiphoton microscopy.," *Biophys J* 92(6), 2212-2222 (2007)

9. O. Nadiarnykh, R. B. LaComb, M. A. Brewer and P. J. Campagnola, "Alterations of the extracellular matrix in ovarian cancer studied by Second Harmonic Generation imaging microscopy," *Bmc Cancer* 10((2010)

10. J. M. Watson, P. F. Rice, S. L. Marion, M. A. Brewer, J. R. Davis, J. J. Rodriguez, U. Utzinger, P. B. Hoyer and J. K. Barton, "Analysis of second-harmonic-generation microscopy in a mouse model of ovarian carcinoma," *Journal of biomedical optics* 17(7), (2012)

11. R. Rezakhaniha, A. Agianniotis, J. T. C. Schrauwen, A. Griffa, D. Sage, C. V. C. Bouten, F. N. van de Vosse, M. Unser and N. Stergiopulos, "Experimental investigation of collagen waviness and orientation in the arterial adventitia using confocal laser scanning microscopy," *Biomechanics and Modeling in Mechanobiology* 11(3-4), 461-473 (2012)

12. P. P. Provenzano, D. R. Inman, K. W. Eliceiri, J. G. Knittel, L. Yan, C. T. Rueden, J. G. White and P. J. Keely, "Collagen density promotes mammary tumor initiation and progression.," *BMC Med* 6(11 (2008)

13. P. P. Provenzano, K. W. Eliceiri, J. M. Campbell, D. R. Inman, J. G. White and P. J. Keely, "Collagen reorganization at the tumor-stromal interface facilitates local invasion.," *BMC Med* 4(1), 38 (2006)

14. G. Falzon, S. Pearson and R. Murison, "Analysis of collagen fibre shape changes in breast cancer," *Physics in Medicine and Biology* 53(23), 6641 (2008)

15. C. Bayan, J. M. Levitt, E. Miller, D. Kaplan and I. Georgakoudi, "Fully automated, quantitative, noninvasive assessment of collagen fiber content and organization in thick collagen gels," *Journal of Applied Physics* 105(10), 102042-102011 (2009)

16. C. Pehlke, J. Doot, K. E. Sung, P. Provenzano, K. Riching, D. Inman, R. Nowak, N. Kouris, B. Ogle, P. Keely, D. Beebe and K. Eliceiri, "Quantification of Collagen Architecture using the Curvelet Transform," (In Press).

17. K. E. Frisch, S. E. Duenwald-Kuehl, H. Kobayashi, C. S. Chamberlain, R. S. Lakes and R. Vanderby, Jr., "Quantification of collagen organization using fractal dimensions and Fourier transforms," *Acta Histochemica* 114(2), (2012)

18. H. Altendorf, E. Decenciere, D. Jeulin, P. D. S. Peixoto, A. Deniset-Besseau, E. Angelini, G. Mosser and M. C. Schanne-Klein, "Imaging and 3D morphological analysis of collagen fibrils," *Journal of Microscopy* 247(2), (2012)

19. W. Hu, H. Li, C. Wang, S. Gou and L. Fu, "Characterization of collagen fibers by means of texture analysis of second harmonic generation images using orientation-dependent gray level co-occurrence matrix method," *Journal of Biomedical Optics* 17(2), (2012)

20. P. P. Provenzano, K. W. Eliceiri and P. J. Keely, "Shining new light on 3D cell motility and the metastatic process.," *Trends Cell Biol* 19(11), 638-648 (2009)

21. J. Wu, B. Rajwa, D. L. Filmer, C. M. Hoffmann, B. Yuan, C. S. Chiang, J. Sturgis and J. P. Robinson, "Analysis of orientations of collagen fibers by novel fiber-tracking software.," *Microsc Microanal* 9(6), 574-580 (2003)

22. J. Wu, B. Rajwa, D. L. Filmer, C. M. Hoffmann, B. Yuan, C. Chiang, J. Sturgis and J. P. Robinson, "Automated quantification and reconstruction of collagen matrix from 3D confocal datasets.," *J Microsc* 210(Pt 2), 158-165 (2003)

23. A. M. Stein, D. A. Vader, L. M. Jawerth, D. A. Weitz and L. M. Sander, "An algorithm for extracting the network geometry of three-dimensional collagen gels.," *J Microsc* 232(3), 463-475 (2008)

24. Z. T. Harmany, R. F. Marcia and R. M. Willett, "This is SPIRAL-TAP: Sparse Poisson Intensity Reconstruction ALgorithms-Theory and Practice," *Ieee Transactions on Image Processing* 21(3), (2012)

25. Y. Sato, S. Nakajima, N. Shiraga, H. Atsumi, S. Yoshida, T. Koller, G. Gerig and R. Kikinis, "Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images," *Medical image analysis* 2(2), 143-168 (1998)

26. J. L. Starck, E. J. Candès and D. L. Donoho, "The curvelet transform for image denoising.," *IEEE Trans Image Process* 11(6), 670-684 (2002)

27. E. Candes, L. Demanet, D. Donoho and L. Ying, "Fast discrete curvelet transforms," *Multiscale Modeling & Simulation* 5(3), (2006)

28. A. Stein, "FIRE FIbeR Extraction," *http://www.ima.umn.edu/~astein/Andrew%20Stein/Software.html* (2008)

29. E. Y. S. Yew and C. J. R. Sheppard, "Effects of axial field components on second harmonic generation microscopy," *Optics Express* 14(3), 1167-1174 (2006)

30. E. Y. S. Yew and C. J. R. Sheppard, "Second harmonic generation polarization microscopy with tightly focused linearly and radially polarized beams," *Optics Communications* 275(2), 453-457 (2007)

31. A. Erikson, J. Oertegren, T. Hompland, C. d. L. Davies and M. Lindgren, "Quantification of the second-order nonlinear susceptibility of collagen I using a laser scanning microscope," *Journal of Biomedical Optics* 12(4), (2007)

32. D. Kedrin, B. Gligorijevic, J. Wyckoff, V. V. Verkhusha, J. Condeelis, J. E. Segall and J. van Rheenen, "Intravital imaging of metastatic behavior through a mammary imaging window," *Nat. Methods* 5(12), 1019-1021 (2008)

33. U. Bal, "Dual tree complex wavelet transform based denoising of optical microscopy images," *Biomedical optics express* 3(12), (2012)

34. A. Beck and M. Teboulle, "Fast Gradient-Based Algorithms for Constrained Total Variation Image Denoising and Deblurring Problems," *Ieee Transactions on Image Processing* 18(11), (2009)

35. J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, J. Y. Tinevez, D. J. White, V. Hartenstein, K. Eliceiri, P. Tomancak and A. Cardona, "Fiji: an open-source platform for biological-image analysis," *Nat. Methods* 9(7), 676-682 (2012)

36. M. H. Longair, D. A. Baker and J. D. Armstrong, "Simple Neurite Tracer: open source software for reconstruction, visualization and analysis of neuronal processes," *Bioinformatics* 27(17), 2453-2454 (2011)

37. E. J. Candes and F. Guo, "New multiscale transforms, minimum total variation synthesis: applications to edge-preserving image reconstruction," *Signal Processing* 82(11), (2002)

38. D. Sage, D. Prodanov, C. Ortiz, J.-Y. Tivenez, J. Pecreaux and J. Schindelin, "A Java package for bi-directional communication and data exchange from Matlab to ImageJ/Fiji."

39. X. Wang, T. Heimann, P. Lo, M. Sumkauskaite, M. Puderbach, M. de Bruijne, H. P. Meinzer and I. Wegner, "Statistical tracking of tree-like tubular structures with efficient branching detection in 3D medical image data," *Physics in Medicine and Biology* 57(16), 18 (2012)

40. V. Mohan, G. Sundaramoorthi and A. Tannenbaum, "Tubular Surface Segmentation for Extracting Anatomical Structures From Medical Imagery," *IEEE Trans. Med. Imaging* 29(12), 1945-1958 (2010)

41. O. Friman, M. Hindennach, C. Kuhnel and H. O. Peitgen, "Multiple hypothesis template tracking of small 3D vessel structures," *Medical Image Analysis* 14(2), 160-171 (2010)

42. J. Ma and G. Plonka, "The Curvelet Transform," *IEEE Signal Processing Magazine* 27(2), (2010)

43. N. R. Pal and S. K. Pal, "A REVIEW ON IMAGE SEGMENTATION TECHNIQUES," *Pattern Recognit.* 26(9), 1277-1294 (1993)

44. I. S. Molchanov and P. Teran, "Distance transforms for real-valued functions," *Journal of Mathematical Analysis and Applications* 278(2), (2003)

45. I. Freund, M. Deutsch and A. Sprecher, "CONNECTIVE-TISSUE POLARITY - OPTICAL 2ND-HARMONIC MICROSCOPY, CROSSED-BEAM SUMMATION, AND SMALL-ANGLE SCATTERING IN RAT-TAIL TENDON," *Biophysical Journal* 50(4), 693-712 (1986)

46. P. Stoller, B. M. Kim, A. M. Rubenchik, K. M. Reiser and L. B. Da Silva, "Polarization-dependent optical second-harmonic imaging of a rat-tail tendon," *J Biomed Opt* 7(2), 205-214 (2002)

47. P. Stoller, K. M. Reiser, P. M. Celliers and A. M. Rubenchik, "Polarization-modulated second harmonic generation in collagen," *Biophysical Journal* 82(6), 3330-3342 (2002)

48. X. Y. Chen, O. Nadiarynkh, S. Plotnikov and P. J. Campagnola, "Second harmonic generation microscopy for quantitative analysis of collagen fibrillar structure," *Nat. Protoc.* 7(4), 654-669 (2012)

49. A. E. Tuer, M. K. Akens, S. Krouglov, D. Sandkuijl, B. C. Wilson, C. M. Whyne and V. Barzda, "Hierarchical Model of Fibrillar Collagen Organization for Interpreting the Second-Order Susceptibility Tensors in Biological Tissue," *Biophysical Journal* 103(10), 2093-2105 (2012)

50. J. Duboisset, D. Ait-Belkacem, M. Roche, H. Rigneault and S. Brasselet, "Generic model of the molecular orientational distribution probed by polarization-resolved second-harmonic generation," *Physical Review A* 85(4), (2012)

51. W. R. Zipfel, R. M. Williams and W. W. Webb, "Nonlinear magic: Multiphoton microscopy in the biosciences," *Nature Biotechnology* 21(11), 1369-1377 (2003)

52. M. W. Conklin, P. P. Provenzano, K. W. Eliceiri, R. Sullivan and P. J. Keely, "Fluorescence lifetime imaging of endogenous fluorophores in histopathology sections reveals differences between normal and tumor epithelium in carcinoma in situ of the breast.," *Cell Biochem Biophys* 53(3), 145-157 (2009)

7 Figure Captions

Figure 1. Representative collagen patterns observed in human breast cancer tissue sections demonstrating the heterogeneous nature of collagen structure. Wavy (A) and straight (B). Dense (C) and well defined (D). Thick bundles (E) and thin strands (F). Discontinuous (G) and continuous (H). Crossing (I) and parallel (J). Scale bar is 10 microns.

Figure 2. Fibers extracted by the FIRE algorithm alone without preprocessing. A and D are the original images, B and E show manual segmentations of the fibers, C and F show the automatic fiber segmentations that are extracted by the FIRE algorithm and show many falsely segmented fibers. Scale bar is 25 microns.

Figure 3. Diagram of the approach for quantitative collagen analysis showing the iterative process for optimizing the performance of a single image-processing filter for fiber tracking. The raw image is processed by the image filter using an initial normalization parameter, the result of which is sent to the FIRE fiber-tracking algorithm. Automated fiber extractions are compared against manually performed

fiber extractions. Several normalization parameters are evaluated and one optimal parameter is selected for each filter based on the fiber evaluation result.

Figure 4. Output of the image processing techniques (row 1) and output of the fiber tracking algorithm (row 2) for a single test case. The first column is without a filter, column 2: GF, column 3: SPTV filter, column 4: TF, and column 5: CT. Scale bar is 25 microns.

Figure 5. Two test cases (A&B), showing different processing methods in each column. The original image (column 1) is shown overlaid with a manual segmentation (column 2), GF (column 3), SPTV (column 4), TF (column 5), and CT filter (column 6) where each filter is followed by FIRE fiber extraction. Scale bar is 25 microns.

Figure 6. F-measure, recall, and precision results comparing the automated segmentation techniques to the manual segmentations of three independent raters, for 25 test cases, representing a total of 9290 fiber evaluations. The error bars indicate the standard deviation between average F-measure, recall and precision scores of each of the raters. Recall is the fraction of relevant fibers that were found. Precision is the fraction of fibers found that were relevant. F-measure is the harmonic sum of recall and precision.

Figure 7. Demonstration of automated fiber segmentation feature extraction in an *in-vivo* mouse model for breast cancer. A mammary window was placed immediately superficial to a palpable mammary tumor and the collagen microenvironment was imaged 8 and 12 weeks of age. Automated fiber extractions are shown overlaid on representative images from the 8 (A) and 12 (B) week time points. The bar graph (C) shows the ratio of the number of wavy fibers to total fibers found in the image. Fibers are labeled wavy if the distance along the fiber divided by the distance between fiber endpoints is greater than 1.08. Error bars indicate one standard deviation of the computed average wavy-fractions among the 9 images analyzed for each time point. Scale bar is 25 microns.

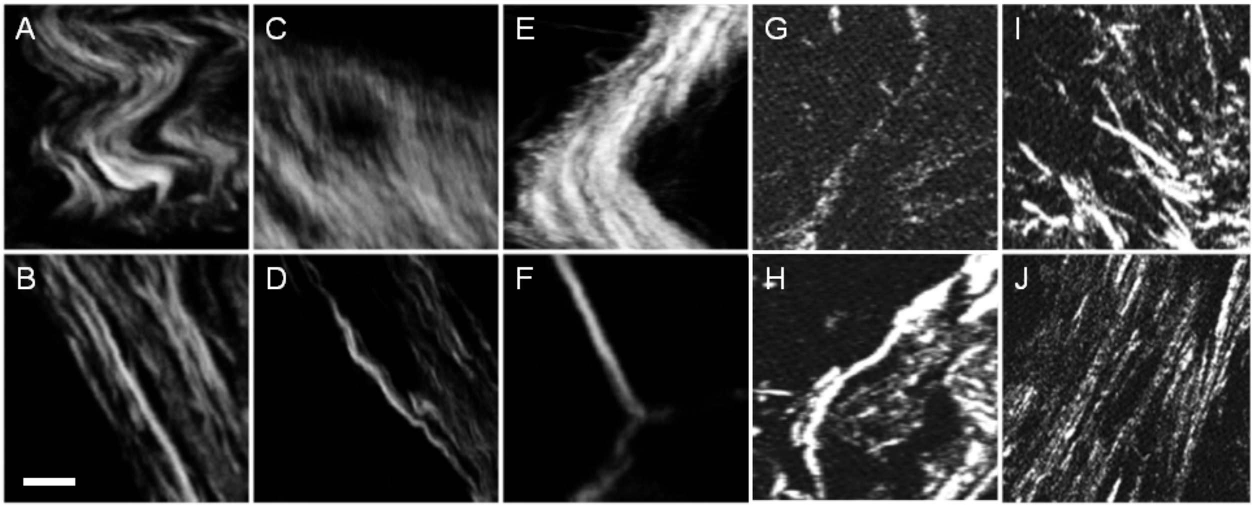


Figure 8. Representative collagen patterns observed in human breast cancer tissue sections demonstrating the heterogeneous nature of collagen structure. Wavy (A) and straight (B). Dense (C) and well defined (D). Thick bundles (E) and thin strands (F). Discontinuous (G) and continuous (H). Crossing (I) and parallel (J). Scale bar is 10 microns.

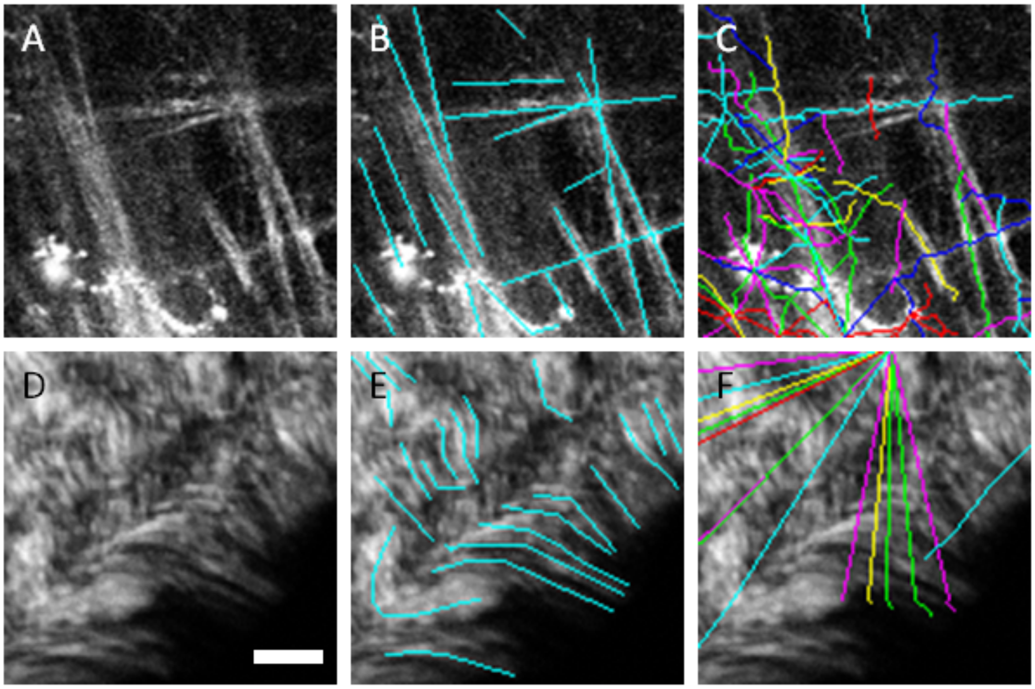


Figure 9. Fibers extracted by the FIRE algorithm alone without preprocessing. A and D are the original images, B and E show manual segmentations of the fibers, C and F show the automatic fiber segmentations that are extracted by the FIRE algorithm and show many falsely segmented fibers. Scale bar is 25 microns.

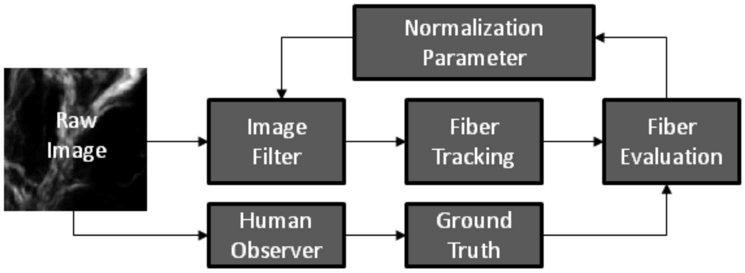


Figure 10. Diagram of the approach for quantitative collagen analysis showing the iterative process for optimizing the performance of a single image-processing filter for fiber tracking. The raw image is processed by the image filter using an initial normalization parameter, the result of which is sent to the FIRE fiber-tracking algorithm. Automated fiber extractions are compared against manually performed fiber extractions. Several normalization parameters are evaluated and one optimal parameter is selected for each filter based on the fiber evaluation result.

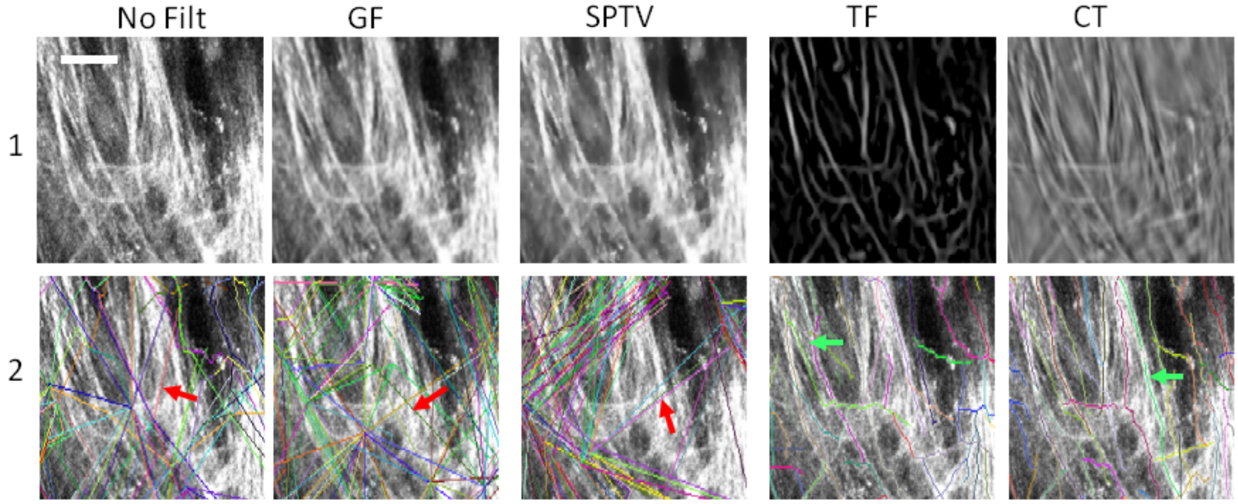


Figure 11. Output of the image processing techniques (row 1) and output of the fiber tracking algorithm (row 2) for a single test case. The first column is without a filter, column 2: GF, column 3: SPTV filter, column 4: TF, and column 5: CT. Scale bar is 25 microns.

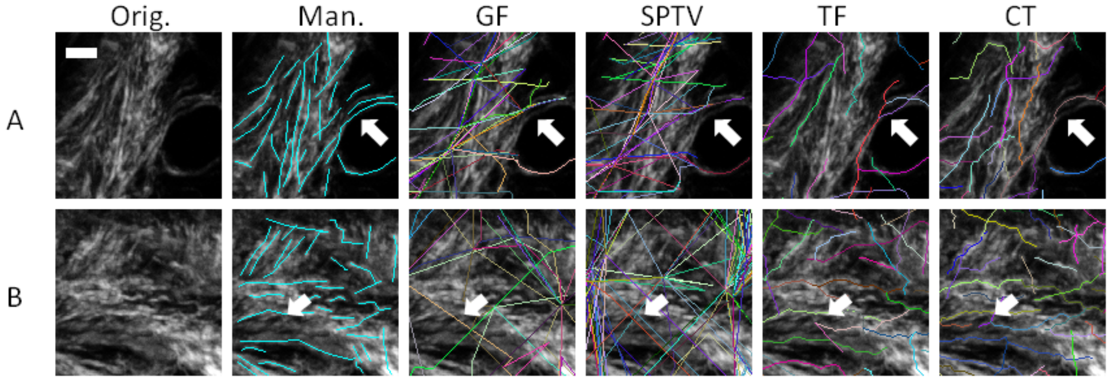


Figure 12. Two test cases (A&B), showing different processing methods in each column. The original image (column 1) is shown overlaid with a manual segmentation (column 2), GF (column 3), SPTV (column 4), TF (column 5), and CT filter (column 6) where each filter is followed by FIRE fiber extraction. Scale bar is 25 microns.

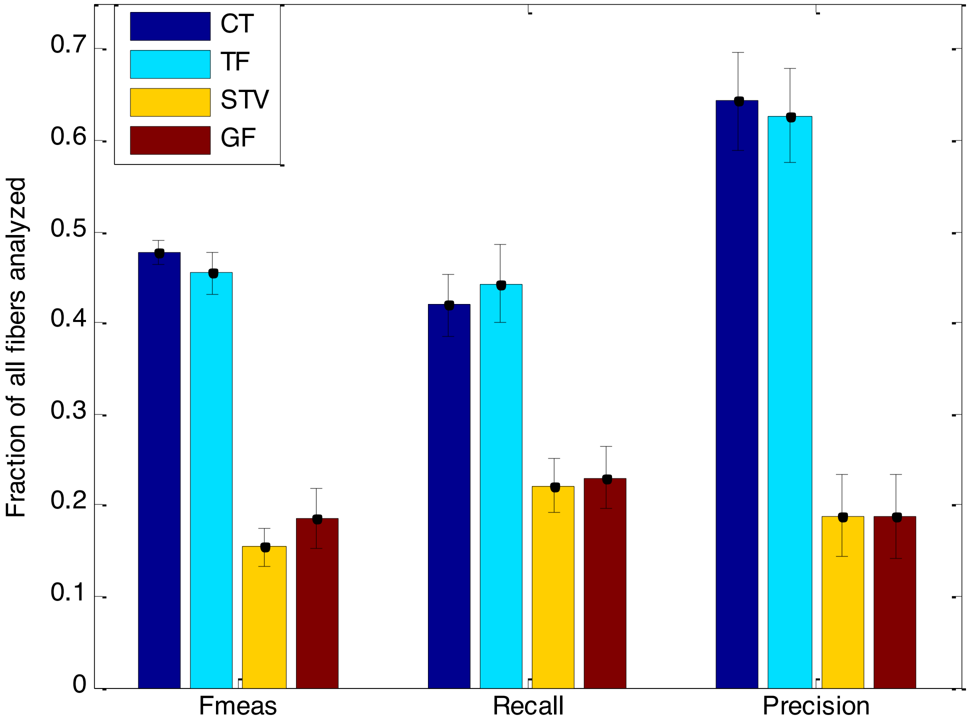


Figure 13. F-measure, recall, and precision results comparing the automated segmentation techniques to the manual segmentations of three independent raters, for 25 test cases, representing a total of 9290 fiber evaluations. The error bars indicate the standard deviation between average F-measure, recall and precision scores of each of the raters. Recall is the fraction of relevant fibers that were found. Precision is the fraction of fibers found that were relevant. F-measure is the harmonic sum of recall and precision.

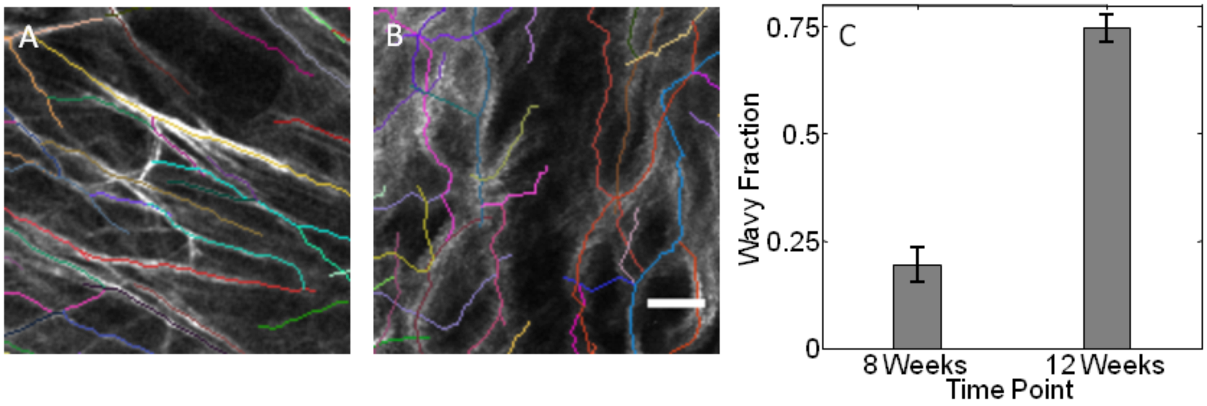


Figure 14. Demonstration of automated fiber segmentation feature extraction in an *in-vivo* mouse model for breast cancer. A mammary window was placed immediately superficial to a palpable mammary tumor and the collagen microenvironment was imaged 8 and 12 weeks of age. Automated fiber extractions are shown overlaid on representative images from the 8 (A) and 12 (B) week time points. The bar graph (C) shows the ratio of the number of wavy fibers to total fibers found in the image. Fibers are labeled wavy if the distance along the fiber divided by the distance between fiber endpoints is greater than 1.08. Error bars indicate one standard deviation of the computed average wavy-fractions among the 9 images analyzed for each time point. Scale bar is 25 microns.

Chapter 4: Automated Quantification of Fiber Topology in Breast Cancer

Adapted from: Automated quantification of aligned collagen for human breast carcinoma prognosis. Jeremy S. Bredfeldt, Yuming Liu, Matthew W. Conklin, Patricia J. Keely, Thomas R. Mackie, Kevin W. Eliceiri . Journal of Pathology Informatics.

Abstract

Background: Mortality in cancer patients is directly attributable to the ability of cancer cells to metastasize to distant sites from the primary tumor. This migration of tumor cells begins with a remodeling of the local tumor microenvironment, including changes to the extracellular matrix and the recruitment of stromal cells, both of which facilitate invasion of tumor cells into the bloodstream. In breast cancer, it has been proposed that the alignment of collagen fibers surrounding tumor epithelial cells can serve as a quantitative image-based biomarker for survival of invasive ductal carcinoma patients. Specific types of collagen alignment have been identified for their prognostic value and now these tumor associated collagen signatures (TACS) are central to several clinical specimen imaging trials. Here, we implement the semi-automated acquisition and analysis of this TACS candidate biomarker and demonstrate a protocol that will allow consistent scoring to be performed throughout large patient cohorts. **Methods:** Using large field of view high resolution microscopy techniques, image processing and supervised learning methods, we are able to quantify and score features of collagen fiber alignment with respect to adjacent tumor-stromal boundaries. **Results:** Our semi-automated technique produced scores that have statistically significant correlation with scores generated by a panel of three human observers. In addition, our system generated classification scores that accurately predicted survival in a cohort of 196 breast cancer patients. Feature rank analysis reveals that TACS positive fibers are more well-aligned with each other, are of generally lower density, and terminate within or near groups of epithelial cells at larger angles of interaction. **Conclusion:** These results demonstrate the utility of a supervised learning protocol for streamlining the analysis of collagen alignment with respect to tumor stromal boundaries.

Key words: Breast cancer, collagen, image processing, machine learning

BACKGROUND

Breast cancer diagnosis and staging have been revolutionized by new molecular screening assays based on immunohistochemistry,[1] fluorescence *in situ* hybridization,[2] and reverse transcription polymerase chain reaction,[3] which are all used to personalize care. These tools are helping patients live longer and receive better treatment than ever before. However, there remains a significant group of breast cancer patients for whom these new techniques ultimately fail, due to several factors including varying patient genotype and primary or acquired resistance to drugs such as the HER2/neu receptor targeting drug trastuzumab (trade name Herceptin).[4] In addition, molecular screens are confounded by the high-degree of intratumor genetic diversity and often require extra tissue sections to be cut, stained and evaluated on top of the standard hematoxylin and eosin (H&E) preparation. New assays that predict patient outcome and response to treatment are therefore critically needed if we are to continue improving breast cancer treatment and prevention. One promising area of development is image based assays, which leverage high content imaging hardware and image analysis software to classify biological samples.[5-7] In many cases, image based analysis does not require more than the standard histopathology H&E stained slides prepared as part of the normal clinical workflow. In this paper, we demonstrate the use of a new image-based assay for predicting patient outcome using information about tumor-stromal interactions from standard H&E stained histopathology specimens.

Aberrant tumor-stromal interactions have been shown to accelerate tumorigenesis in breast cancer.[8-10] The importance of stromal collagen in breast cancer is highlighted by the link between breast cancer, breast density, and the increased deposition of stromal collagen.[11-15] Interestingly, although mammographic density, which is attributable to collagen content, is one of the largest risk factors for the development of breast tumors, there is currently no clinical intervention based on mammographic density alone. This is due in part to the lack of a clear correlation observed between increased mammographic density and patient outcome. Most of the work to date[16-19] has defined mammographic density as a etiological factor and not as a prognostic factor. Recently, Cil *et al.*[20] explored mammographic density as predictor of local breast cancer recurrence. They reported that women with intermediate and high breast density had a significantly elevated risk to develop a local breast cancer recurrence. However, follow-up clinical trials that incorporate additional risk factors such as obesity are needed to examine the possible prognostic value of mammographic density in large and diverse patient cohorts before using density as a possible clinical target. As well recently there has been an effort to investigate the underlying contributor to mammographic density, focusing on one of the largest components present in the dense stroma, collagen. Several studies have shown a link between collagen remodeling and the invasion and progression of mammary cancer in mouse models.[21-23] Furthermore, there was a link observed between collagen morphology, particularly collagen alignment, and breast cancer patient outcome.[24] Provenzano *et al.*[21] first introduced the so called tumor associated collagen signature (TACS) nomenclature to describe collagen alignment patterns. The TACS phenotypes are currently classified into three groups. TACS-1 describes the standard desmoplastic response of increased collagen deposition surrounding initiating tumor cells. TACS-2 is observed as straightened fibers aligned tangentially around developing tumors, while TAC-3 is seen as radially aligned fibers that facilitate local invasion.[25] Conklin *et al*.[24] qualitatively searched for these patterns in human breast cancer samples and through extensive manual analysis found that the presence of the TACS-3 alignment phenotype was a prognostic indicator for disease free and disease specific survival (DFS and DSS respectively) for invasive breast cancer patients. Our quantitative study, presented here, computationally builds on this previous work by defining an algorithmic model for TACS-3 and applying this model to the same cohort of patients.[24]

Previous collagen alignment studies have largely been facilitated by the development of second harmonic generation (SHG) microscopy techniques, which have the ability to capture high contrast images of the collagen fiber extracellular matrix without the need for exogenous stains.[26-29] The application of SHG imaging in cancer research is growing rapidly. For example, changes in the ratio of the forward SHG (F_{SHG}) to backward propagating SHG signal have been recently linked to breast tumor progression[30] and positive lymph node status.[31] SHG directionality was also used by Ajeti *et al.* to quantify the collagen composition in breast cancer models,[32] while Ambekar *et al.* used Fourier transform and polarization-resolved SHG imaging to differentiate malignant from benign tissues in breast biopsies.[33]

In addition, many new computational techniques are being developed to quantify patterns observed in SHG images. For example, a directional gradient method developed by Altendorf *et al*.[34] provides three-dimensional orientation and radius information about fibers in SHG images. Due to the fibrous nature of the collagen matrix, SHG images are particularly well-suited for the curvelet transform (CT), which is a multiscale, orientation sensitive version of the wavelet transform. The CT[35] and combined fiber tracking methods[36] have been applied to extract fiber orientation, length, curvature and radius from SHG images of collagen. One key feature however that is missing from all of the available image analysis techniques is the ability to incorporate cellular information into the analysis. The interaction between tumor cells and collagen fibers cannot be fully assessed without integration of information about cellular morphology and associated collagen morphology. As well this information is critical for finding regions of interest (ROI) with TACS, an essential task for any type of high-throughput screening where manual searching is not practical. Herein, we describe a computational protocol that achieves this goal by integrating information about collagen fibers from SHG images with information about cells captured through bright field imaging of standard H&E stained slides to perform highly automated, prognostic TACS-3 scoring.

In order for TACS to become a useful and fully validated biomarker, it must be screened for in several large studies containing many patients and diverse populations. In addition, besides screening in heterogeneous populations, it ideally needs to be screened in diverse sample types to account for possible subtle differences in surgery, pathology or sample preparation that could negatively impact sample consistency. This ability to rapidly screen in many sample types of large diverse populations would also open the door for TACS to be explored in other cancer types such as pancreatic and renal cancer. Heretofore, there has not been a method that automates enough of the process to enable such large scale adaptation. In previous studies, collagen fiber angles were measured by hand, one at a time, using ImageJ ROI marking tools.[21,23] These experiments used information gathered *a priori* or from autofluorescence to identify tumor-stromal boundaries. In addition, imaging locations were chosen manually. Conklin *et al*. manually captured each individual image, used bright field images to manually identify tumor-stromal boundaries, and manually estimated collagen fiber angles.[24] In each of these cases, many subjective decisions were made while identifying which areas to image, which fibers to measure and what should be considered a tumor-stromal boundary. There has been progress made in automating the fiber angle analysis steps of this task.[35,36,37-39] However, none of these methods can automate all four steps of the TACS analysis process, which are: (1) Image capture, (2) fiber angle measurement, (3) tumor-stromal boundary identification, and (4) relative angle measurement between fiber and boundary. In this paper, we use image analysis and supervised learning techniques to enable the automation of each of these tasks. The block diagram of our imaging and analysis protocol is shown in Figure 1. Starting with the previously-imaged invasive breast cancer tissue microarray (TMA), we captured registered, whole-slide SHG and bright field images, extracted fibers from the SHG images, identified tumor-stromal boundaries from the bright field images, and measured relative angles, all in a scripted pipelined process that requires little human intervention. We believe that this method will allow significantly larger scale studies to be performed in order to validate TACS-3 as a prognostic biomarker in breast cancer and potentially other cancer types, and to investigate if TACS-3 can be used to predict patient response to targeted therapies.

METHODS

Human Breast Carcinoma Tissue Microarray

The TMA used here was the same as that used by Conklin *et al.*[24] for the manual collagen alignment analysis. The clinical profiles of all patients whose tissue was included in this TMA have been described in a previous study.[39] All tissue and patient information used in this study were acquired following Institutional Review Board approval. Tumor tissues from 353 patients diagnosed with invasive carcinoma were resected by the same surgeon between 1981 and 1995.

Pieces of each resected tumor were embedded in paraffin according to standard histopathology protocols. After tumors smaller than 5 mm and severely damaged samples were excluded, 196 patients remained for analysis. Sections of 4 μm thickness were cut from archived TMA blocks containing 1.0 mm diameter tissue cores, placed on glass slides, stained with H&E and mounted under a glass coverslip. Patients were followed for a median of 6.2 years, ranging from 1 month to 18.6 years.

Imaging System

All samples in this study were imaged with the custom built integrated F_{SHG}/bright field imaging system shown in Figure 2. A MIRA 900 Ti:Sapphire laser (Coherent, Santa Clara, CA) tuned to 780 nm, with a pulse length of approximately 100 fs, was directed through a Pockel's cell (ConOptics, Danbury, CT), half and quarter waveplates (ThorLabs, Newton, NJ), beam expander (ThorLabs), a 3 mm galvanometer driven mirror pair (Cambridge, Bedford, MA), a scan/tube lens pair (ThorLabs), through a dichroic beam splitter (Semrock, Rochester, NY) and focused by a 20X/0.75NA objective (Nikon, Melville, NY). SHG light was collected in the forward direction with a 0.54 NA condenser (ThorLabs) and filtered with an interference filter centered at 390 nm with a full width at half maximum bandwidth of 22.4 nm (Semrock). The back aperture of the condenser lens was imaged onto the 5 mm aperture of a 7422–40P photomultiplier tube (Hamamatsu, Hamamatsu, Japan) the signal from which was amplified with a C7319 integrating amplifier (Hamamatsu) and sampled with an analog to digital converter (Innovative Integration, Simi Valley, CA). Timing between the galvo scanners, signal acquisition, and motorized stage positioning was achieved using our custom software called WiscScan.[41] The Rapid Automated Modular Microscope system (Applied Scientific Instrumentation, Eugene, OR) served as our microscope base and we used ASI motorized translation stages for *x*, *y*, and *z* motion control. The SHG light source was verified to be circularly polarized at the sample using the protocol of Chen *et al*.[29] SHG images were captured as stacks of three images spaced 3 μm apart, then z-projected to improve field flatness. Bright field images were captured with the same system using a MCWHL2 white LED lamp (ThorLabs) set up for Kohler illumination. White light from this lamp was separated from SHG light traveling through the condenser assembly using a short pass dichroic mirror with a cutoff at 670 nm (Semrock). An RGB camera (QImaging, Surrey, BC, Canada) was used to capture bright field images through WiscScan to allow for acquisition within a single application. Both SHG and white light images were tiled with 10% overlap using automation provided by WiscScan. Stage positions for individual images and pixel size data were stored in Bio-Formats image metadata[42] and this was then used by the grid/collection stitching ImageJ plugin[43] to reassemble a high-resolution large field of view image of the entire TMA. When capturing large field of view images, the sample plane often walks out of the in-focus imaging plane as the stage is translated over large distances in *x* or *y*. We alleviated this issue using the Continuous Reflection Interface Sampling and Positioning autofocus system (Applied Scientific Instrumentation), which maintained an accurate distance between the coverslip and the objective throughout the whole slide stitched image capture. This allowed for a single bright field image to be captured at each location rather than a z-stack, improving capture speed, reconstruction speed, and reducing production of unnecessary data. After SHG and white light images were captured and stitched, the two modalities were manually registered with the landmark correspondences ImageJ plugin using five control points per image. The image of the entire TMA was registered in a single step, and then each individual TMA core was cropped out of the full TMA image, producing 196 images. The resulting TMA core images were each 2048 × 2048 pixels, consisting of four eight-bit channels. The first three channels represented the red, green and blue planes of the white light image, while the fourth channel contained the SHG information.

<H2>Tumor associated collagen signatures (TACS)-3 Model

Our TACS-3 model was based on previously published observations relating collagen structure to breast cancer progression and survival. In these studies, the first step in the TACS-3 scoring process was the identification of groups of straightened, aligned collagen fibers. The second step was to determine if those fibers terminate at or near regions of epithelial cells at steep angles. If a fiber met both of these criteria, then it was considered TACS-3 positive. If one or more TACS-3 positive fibers were found in a sub-region of an image, then that region was scored TACS-3 positive. The number of regions with TACS-3 positive scores was then used to score the entire image. There were many details in these steps and defining parameters to account for each step would have produced a potentially fragile model. Instead, we have implemented a supervised learning approach that allows the data to most appropriately define the model. We performed this task computationally using a series of cascaded classifiers. The first classifier was trained to find epithelial regions in the images using a small training set of annotated ROIs. The resultant epithelial cell model was then used to segment epithelial cell regions within the entire cohort of images. Features describing the epithelial regions were then combined with features derived from our fiber extraction algorithm and were fed into a second classifier, which was trained to score each image as being TACS-3 positive or negative based on a training set of annotated images. TACS-3 scores were then fed into a cox proportional hazard model to regress to censored survival data.

Our entire image cohort can be represented by a set S of registered SHG and bright field (RGB) images. Each image $I_i(\overset{\vee}{u}, g) \in S$ was composed of pixels $\overset{\vee}{u} = (x_i, y_i) \in R^2, 0 < x_i, y_i < N$ and the function $g(\overset{\vee}{u})$ which mapped each pixel to a quartet of intensity values corresponding to the R, G, B, and SHG intensity channels of the image. The first step of our protocol involved the extraction of collagen fiber objects from the SHG channel. Next, epithelial cell clusters were segmented from the R, G, and B channels. Each fiber from the first step was then associated with epithelial cell clusters from the second step to create a feature set for each fiber. Average feature values for all fibers in the image were then used to classify images based on a training set of images.

Fiber Extraction

We applied a technique called CT-FIRE[36] to the SHG images to enhance, trace and extract a network of collagen fibers for each SHG image $I(\check{u}, g_{shg})$. CT-FIRE combines the advantage of the CT[44] for denoising the image and enhancing the fiber ridge features with the advantage of a fiber tracing algorithm[45] for automatic fiber extraction, being capable of extracting fiber geometric information such as length, angle, width, and curvature of each fiber. We applied the fast discrete CT (FDCT) to capture a collection of coefficients C^D in curvelet space, which are defined as the inner product of the input SHG image channel with each of the curvelet basis functions

$$C^D(j,l,k) = \sum_{0<x_i, y_{i<n}} I(\check{u}, g_{shg})\, \gamma^D_{jkl}(\check{u})$$

where $\gamma^D_{jkl}(\check{u})$ is the digital curvelet waveform and jkl represent the scale, orientation, and location indices, respectively. We used the open source FDCT MATLAB[44] library and specifically the "wrapping" version of the FDCT due to its simplicity. To denoise the image, we set all curvelet coefficients to zero that fall below a user defined threshold T as shown below

$$C^D_T(j,l,k) = \begin{cases} C^D(j,l,k), & \left| C^D(j,l,k) \right| > T \\ 0, & otherwise \end{cases}$$

This threshold was determined empirically on a small subset of SHG images to determine the appropriate level of noise reduction. The inverse FDCT was then applied to reconstruct an edge enhanced, noise reduced version of the SHG image. After reconstruction, CT-FIRE traced fibers, using the method of Stein *et al.*,[45] by first finding local maxima in the result of the smoothed distance transform. The distance transform computed the distance from each foreground pixel to the nearest background pixel. Fiber branches were formed by creating regions surrounding each local maxima, the size of which were defined by the result of the distance transform at the location of the local maximum point. The edges of this region were then searched for further local maxima. This process was repeated until no new local maxima were found indicating the end of a fiber branch. Short branches were then pruned from the network and closely spaced, similarly oriented fibers were merged. Fiber width (FW) was quantified for each extracted fiber by averaging the fiber widths $(2R_i)$ at n points that were used to form the fiber

$$FW = \frac{1}{n}\sum_{i=1}^{n} 2R_i$$

Where R_i is the fiber radius at the i^{th} point, estimated by the result of the distance transform at that location. Fiber straightness (FS) was quantified for each extracted fiber by dividing the distance between the end points of the fiber $(d_n)$by the distance along the path of the fiber (d_0)

$$FS = \frac{d_n}{d_0}$$

Thus for perfectly straight fibers $FS = 1.0$ and wavy fibers$FS < 1.0$. After fiber objects have been extracted from each of the images, we next segment epithelial cell regions.

Epithelial Cell Segmentation

The TACS-3 phenotype consists of straightened aligned collagen fibers that terminate near regions of epithelial cells such that the angles of the collagen fibers appear perpendicular to the epithelial stromal boundary. Detecting this TACS-3 phenotype requires knowledge of the locations of epithelial cells within the sample. We must then identify regions of epithelial cell clusters and identify a boundary between the epithelial cells and surrounding stroma. This task was performed in two steps outlined here. Step 1 used the Trainable Weka Segmentation ImageJ plugin[46] to find epithelial cell nuclei and step 2 applied a cascaded matched filter, threshold operation to identify clusters and boundaries. The details of these steps are given below.

For step 1, a training set of 15 cropped 256 × 256 pixel images denoted as $t_i(\vec{u}_t, g) \in S$ was created that contains representative features from five classes: epithelial cell nuclei, other cell nuclei (including lymphocytes and fibroblasts), cytoplasm, collagen, and background. A further subgroup of pixels within the training images $\vec{u}_a \in \vec{u}_t$ were annotated as belonging to each class $w_k(\vec{u}_a), k \in (1,...,5)$. A feature vector $p_i(\vec{u}_t, g)$ was computed for each pixel and each channel of the training image where $i \in (1....d)$ was the feature index and d was the dimensionality of the feature subspace. The feature set we used is listed in Table 1 and incorporates features at five scales for a total of 80 feature planes for each image channel. The detailed implementations for each of these features are given in the online documentation for the Weka Segmentation plugin.[47] We then used a multithreaded implementation of the random forests classifier[48,49] with a forest of 200 trees and two random features per node to build a model based on $\vec{u}_a$ and w_k. The trained model was then applied to every pixel $\vec{u}$ in the cohort producing a probability map for each class and each image using a scripted version of the plugin.

For the second step in the segmentation process, the epithelial class probability map was filtered with a Gaussian filter matched to the average width of the epithelial cell nuclei (three microns) and thresholded such that the top 80% of resulting pixels were retained. The resulting image was then filtered with a Gaussian filter matched to the width of the average sized epithelial cell cluster (25 microns), then finally thresholded such that, again, the top 80% of resulting pixels were retained. Following the final threshold step, regions smaller than 50 pixels in area were discarded and a mask was generated with epithelial cell clusters in the foreground and all else in the background. Epithelial mask pixels are represented here as e_i while epithelial region boundary pixels were created using an eight-connected neighborhood and are denoted as b_i. Mask images were saved as tiff files and read, along with the extracted fiber data, into the CurveAlign software, described more below, for fiber/epithelial region feature extraction. Outlines of the resulting mask files were overlaid onto the original white light images to qualitatively validate the segmentation accuracy of the applied epithelial region model.

Combined Fiber-Epithelial Features and Fiber Classification

In the sections above, we described our methods for epithelial cluster segmentation and collagen fiber extraction. With these two pieces of information, we associated fibers with epithelial cell clusters and measured the interaction between the two using the features described here. This task was performed by an open source, MATLAB based tool called CurveAlign.[35] This tool started by reading in a fiber database file (generated by CT-FIRE) and an epithelial mask file (generated by our epithelial segmentation script). A feature vector p_i was then built for each fiber endpoint $v_i \in R^2$ in the image. The feature vector was populated directly with features derived above in the fiber extraction section including fiber length, curvature, radius and grey level. Both endpoints were given the same values for these single fiber derived features. The rest of the features were unique to each fiber end point. All features used in TACS-3 fiber classification are listed in Table 1. Many of the features in this section rely heavily on the nearest neighbor search routine which is formulated here as

$$\varphi^n(X,Q) = \underset{X \in D}{\arg\min}\, \rho(X,Q)$$

Where $D = \{X_1, \ldots, X_n\}$ is a set of vectors in R^2, Q is a query vector, $\rho(X,Q)$ is the Euclidean norm $\|X - Q\|_2$, and the result $\varphi^n(X,Q)$ is a vector of n points in X that are nearest to each point in Q. Given a collection of points on a two-dimensional plane, if we select a query point, this algorithm will return the nearest neighbors within our collection of points. For example, a fiber end point can be used as the query point and we can use this algorithm to search for the nearest point in the list of epithelial cell boundary points, as described below. In addition, some of the features compute a metric for alignment using vector addition according to the following algorithm

$$\sigma(Q_X) = \frac{1}{n}\left|\sum_X \exp(i \times \theta)\right|$$

where $\theta \in \{0,...,\pi\}$ is a vector of n orientations associated with the vector of n positions in X. The factor of two is included since we used fiber orientations supported from $0\, to\, \pi$ rather than full $0\, to\, 2\pi$ oriented direction vectors. In words, the alignment metric is calculated as the normalized vector sum of orientation vectors. The larger the vector sum, the more aligned the group of fibers. On the other hand, if the vector sum is small, then the group of fibers is more randomly oriented. Fiber density features were computed as the average distance from the current fiber endpoint to the n =2, 4, 8, and 16 nearest neighbors. The density features for fiber endpoint v are therefore given by:

$$fd_{n,v} = \frac{1}{n}\sum_j \rho(v - \vartheta_j^n(v_i, v_j))$$

If fiber density is higher, then this result will be lower, since it is measuring the average distance between fibers. Fiber alignment features were computed as the absolute values of the vector sum of the n nearest neighbor fiber endpoints and are given by $fa_{n,v} = \sigma([\theta_v, \theta_{\vartheta_j^n(v_{i,v})}])$ for fiber endpoint v. In the equation for $fa_{n,v}$,$[\cdot,\cdot]$ indicates vector concatenation and n = 2, 4, 8, and 16.

Features that incorporate epithelial cell information included distance to nearest epithelial mask point $de_v = \rho(v - \varphi^1(e_i, v))$ and distance to nearest epithelial region boundary $db_v = \rho(v - \varphi^1(b_i, v))$. These features had the same value if the fiber end point was outside an epithelial cell region; however, they were different if the end point was colocal with an epithelial region. Next, we extracted relative angle features. Angle with respect to nearest epithelial region boundary point was computed as $ab_v = \sigma([\theta_v, \theta_{\varphi(b_{i,v})}])$, and angle with respect to nearest "extension boundary intersection point" was given by $ae_v^r = \sigma([\theta_v, \theta_{\varphi(l_i^r, v)}])$, where the set of points in l^r was computed by taking the intersection of all epithelial boundary points b_i and a line of length $2r$ extending from the fiber endpoint at an angle θ and is formulated below

$$l^r = b_i \cap bres(v - q, v + q)$$

where $bres$ indicates a modified Bresenham algorithm[50] which is used to find all pixels along a line between two points. The term $q = [r \exp(\theta_v)]$ is the offset from v in the x and y directions. For this last feature, three values of r (50, 100 and 200 μm) were calculated. These three lengths corresponded to 5, 10, and 20 times the diameter of a typical epithelial cell and were selected based on estimates of intercellular signaling distances.[51] If no intersection was found, then the ae_v^r feature value was set to zero. The angle of the tumor stromal boundary line $\theta_{\varphi(l_i^r, v)}$ was estimated by fitting a quadratic to nine contiguous points on the boundary surrounding the intersection point (or nearest boundary point in the previous feature) and computing the tangent angle of the line fit at the midpoint. The steps in the process of relative angle feature extraction are diagrammed in Figure 3.

Each of these fiber level features p_i were calculated for every fiber endpoint v_i in the cohort. Fiber level features were then averaged among all fibers in a given image and training was performed with a subset of 16 images $I_t \in I_i$ that had been manually annotated as being TACS-3 positive or negative. A linear support vector machine (SVM) was used to build a model, which was then applied to all 196 images in the cohort for classifying each image as being TACS-3 positive or negative.

Classification and Survival Analysis

The TACS-3 scores were correlated with DFS and DSS data using the Cox-proportional-hazards regression method.[52] DFS was defined as the time from date of diagnosis to the first date of recurrence and DSS was defined as the time from diagnosis to death from breast cancer or date of last follow-up evaluation. In both cases, all other events were censored. The Kaplan-Meier method was used to compare DFS and DSS between TACS-3 negative and TACS-3 positive patients. Hazard ratios were computed using a log-rank test. Correlations between manual and computationally generated TACS-3 scores were made using the Pearson's linear correlation coefficient.

RESULTS

Registered SHG and bright field images of a subsample of the TMA are shown in Figure 4 along with two zoomed versions of regions within the image. SHG information is added as an alpha channel on top of the raw RGB bright field image and pseudo colored yellow. The fully zoomed panel shows the detail available in the full resolution images captured with the 20X, 0.75 NA lens and shows a region with a positive TACS-3 signature. A collection of three more TACS-3 positive and three TACS-3 negative regions were cropped out of the TMA images and shown in

Figure 5. These images illustrate the features that are common to the TACS-3 signature including straightened, aligned fibers terminating in or near regions of epithelial cells at near perpendicular angles with respect to the epithelial region border. In addition, the TACS-3 negative cases show wavy fibers, fibers that terminate at adipose tissue, and a curved fiber encapsulating an epithelial cell cluster [Figure 5d-f, respectively].

A sample of our fiber extraction and epithelial region segmentation results are shown in Figures 6 and 7, respectively. In both cases, epithelial region segmentation and fiber extraction were observed to accurately represent the data. The orientations of the epithelial cell region boundaries were compared to collagen fiber angles derived from the results of a fiber object extraction algorithm CT-FIRE which has been shown to perform well in comparison to other techniques.[36] A representative sample of the results produced by this algorithm are shown in Figure 6. The intermediate product after the CT denoising step is shown in Figure 6a, while the extracted fiber network is shown overlaid on the original SHG image as shown in Figure 6b. Although some fibers are over-or under-segmented (annotated by green arrows), most of the extracted fibers properly represent the data. Figure 7 clearly demonstrates the ability of our epithelial cell segmentation algorithm to properly classify many of the regions of epithelial cells as positive. However, a few small regions of stromal fibroblasts and endothelial cells are included in the epithelial cell regions (annotated by green arrows). Although these errors occurred occasionally throughout the cohort, the noise they generated did not overcome the TACS-3 signal. Another feature evident in Figure 7d is the smoothness of the epithelial region boundaries. The boundary smoothness was dependent on the selection of our filter widths and binary mask thresholds. These parameters were selected to accurately represent the boundary orientation at the spatial scale of the epithelial cell regions.

Although correlation with survival is our ultimate goal, automated TACS-3 scores should also correlate with manual scores for each of the images. The Pearson linear correlation coefficient was used to determine this correlation, the results of which are tabulated in Table 2. The manual analysis performed by Conklin *et al*. produced three scores. Score 1 was the number of TACS-3 positive regions divided by the total number of regions analyzed, score 2 was the average number of TACS-3 positive votes per region among three observers, and score 3 indicated if one or more region received a TACS-3 positive rating. Table 2 shows positive correlation between all manual scoring methods and our computational scoring system presented here, with the highest correlation observed to be with manual score 2.

The Kaplan-Meier curves in Figure 8 demonstrate the prognostic potential of our TACS-3 scoring system. TACS-3 negative patients showed significantly better disease-free and disease-specific survival compared to TACS-3 positive patients. In addition, Cox proportional hazard regression showed significant correlation between our computationally generated TACS-3 scores and survival as listed in Table 3. We also correlated scores created by individual fiber feature metrics with survival. Although fiber features alone were correlated with survival, the highest correlation was observed when the TACS-3 scores were composed of multiple integrated fiber/epithelial features. This result shows that a multimodality imaging and analysis approach that combines features of not only collagen fibers, but both collagen fibers and cellular structures is most likely to succeed in predicting survival.

Table 4 lists the 14 most informative features in the TACS-3 scoring process ranked according to their weight produced by the linear SVM algorithm. The SVM weight was used to assess, which features were more or less informative in the classification. Of particular interest are the features labeled as “nearest distance to boundary” and “inside epithelial region”. These features indicate the proximity between fibers and epithelial cell regions and were highly important in the TACS-3 classification. In addition, the difference in mean feature scores $d_f = f_p - f_n$ for the training set is shown in Table 4 for each of the ranked features. If d_f is >0, then the TACS-3 positive images had larger values for those features and if d_f is <0, then the TACS-3 negative images had larger values. For example, the density features resulted in lower d_f values in the TACS-3 positive cases indicating that the TACS-3 positive images had lower density collagen fibers. On the other hand, d_f was positive for the alignment features indicating that TACS-3 positive images tended to have more aligned fibers. Interestingly, relative boundary angle was not as highly informative as many other features; however, still was ranked within the top 14 of 27 features.

DISCUSSION AND CONCLUSIONS

The search for new prognostic and predictive breast cancer biomarkers is motivated by the need to improve patient outcome. A significant number of patients present with none of the currently available markers. In addition, survival and treatment response is often heterogeneous among patients within current biomarker classifications. The discovery and validation of new biomarkers will help to further improve breast cancer diagnosis and treatment planning. These new biomarkers need to be quantifiable, scalable and ideally correlate with both disease outcome and treatment specific response. The candidate biomarker we are focused on in this study (TACS-3) measures collagen alignment relative to tumor-stromal boundaries and has been associated with progression in mouse models and has been shown to predict disease recurrence and survival in human patients. Here, we demonstrate a protocol for using large field of view imaging techniques, image analysis and supervised learning to automate and quantify all of the steps in the process of TACS-3 scoring. These advances provide the tools for increasing the scale of TACS-3 investigations and applying TACS-3 scoring to cancers in other tissues such as ovarian[53] and pancreatic cancer[54,55] where collagen fiber characteristics are predicted to correlate with prognosis. These techniques could also be used to characterize other TACS both current and yet to be identified to see if they have research value in animal models or prognostic value in clinical specimens.

Tumor associated collagen signature (TACS)-3 analysis requires the simultaneous analysis of information about epithelial cells and extracellular collagen. The interactions between collagen and cells can only be assessed computationally if the cellular information is carefully registered with images of the collagen. We have therefore optimized our imaging system for highly automated capture of large fields of view, registered SHG and bright field images of stained microscope slides with the purpose of analyzing collagen angle with respect to cell cluster boundaries. For this paper, we originally planned to use the same SHG and bright field images captured by Conklin *et al*.[24] since these were already manually annotated. Unfortunately, these images contained artifacts, which, although trivial for the human visual perception system to overcome, were extremely difficult for our computational systems to handle effectively. For example, SHG images were originally captured in the backwards direction with elliptically polarized light, causing two artifacts. The first was simply a low signal to noise level due to few SHG photons traveling in the backward direction from the thin tissue sections.[56,57] The second artifact was observed as a larger relative SHG signal from fibers in the direction parallel to the long axis of the laser polarization ellipse.[58] Artifacts in the bright field images included significant vignetting at field edges and low signal to noise due to short exposure times. These artifacts were easily hurdled by the human observers making TACS-3 assessments in a previous study.[24] However, they are particularly difficult to handle by a computer vision based approach. We therefore decided to develop an optimized imaging system and protocol that would fix many of these artifacts and allow for more consistent automated imaging. Similar image quality and consistency can be achieved with other SHG microscopes including commercial systems with the appropriate hardware, but our analysis protocol did identify a necessary rigorous acquisition protocol that is best achieved with our new automated SHG microscope. In general, the system should allow for F_{SHG} and bright field imaging with a field of view as large and flat as possible, numerical aperture of at least 0.75, automated xyz motion control with appropriate position logging, circular polarization at the sample for SHG imaging, autofocus and automated switching between SHG and bright field imaging.

Our system of imaging and analysis to produce prognostic TACS-3 scores uses standard histopathology H&E slide preparations. The technique is therefore completely compatible with routine clinical protocols and is intended to augment currently available diagnostic tests. The current process requires no changes to current clinical protocol and the sample is returned to the clinician unmodified. We present a system that uses SHG imaging to capture collagen fiber images; however, wide field polarization sensitive techniques[59] such as LC-PolScope[60] or Picrosirius red staining[61,62] might be used to alternatively capture images of collagen fibers. One advantage of using SHG is that it does not require additional stains and can capture three-dimensional fiber information in thick, unstained tissue samples. Unfortunately, when imaging in thick unstained tissue, the identification of epithelial regions can be difficult; however, techniques using autofluoresence and fluorescent lifetime imaging have been shown to be capable of this task.[63,64] As implemented here, our TACS-3 scoring algorithm is necessarily two-dimensional, since we are relying on H&E stained slides for our epithelial cell information. However, fiber extraction, epithelial region segmentation and relative angle measurements can be extended to three-dimensions without significant alteration of our general protocol. In addition, although our current TACS-3 scoring protocol is able to process standard H&E stains, staining for epithelial cells, with, for example, pan-cytokeratin conjugated stains, may simplify and improve epithelial cell segmentation. Future methods may also be adapted to segment clusters of fibroblasts, macrophages and other stromal cells, whose proximity and relative morphological structure with respect to surrounding collagen fibers may further improve correlation with survival or metastatic potential.

Collagen alignment related image features are interesting not only because they have been shown to be prognostic, but because they have been shown to be directly linked to cancer biology. Researchers have found that cells are more likely to invade along parallel, aligned collagen fibers,[25,65] features that are directly being measured by our system. Access to the breadth of fiber data available with our techniques could lead to advances in our understanding of these biological phenomena. Relevant feature sets are not always available with other machine vision systems developed for biological image classification. For example, although WNDCHRM[5] is an extremely powerful image classification tool, informative image features often do not relate to the biology at hand. In the case of our TACS-3 analysis system, biological observations have driven the image analysis model; therefore, features are more easily linked back to biological functions potentially revealing new insights.

High mammographic density is one of the largest risk factors for the development of breast cancer and has been associated with increased epithelial cellularity and increased collagen density.[12,14,19] Increased collagen density has been observed to promote tumor progression in a mouse tumor model[23] and in node positive breast cancer[31] leading one to potentially conclude that collagen density causes elevated risk. However, Maller *et al*.[66] observed that high density, nonfibrillar collagen protected against tumor progression and alternatively, that linearized collagen fibers induced invasive cellular behavior. In agreement with these recent findings, we observe here that TACS-3 fibers are more commonly present in regions of lower fiber density and are more likely to be thinner, more linearized fibers. Thick, curvy, and denser collagen fibers are unlikely to contain TACS-3 fibers and are observed to be associated with a better prognosis. These observations support the hypothesis that collagen fiber shape and organization is a key aspect of the invasive extracellular matrix (ECM) phenotype.

The imaging instrumentation presented here consists of a relatively compact and highly automated multiphoton microscope with an integrated bright field slide scanner. The system has been optimized to capture registered whole slide images of both bright field and SHG images of histopathology slides by imaging each small ×20 field of view and automatically aligning and stitching each image together. Capturing large fields of view in this manner allows for a more thorough and consistent data collection potentially reducing sampling bias and supporting pipelined computational image analysis. The registration of cellular with extracellular collagen information provided by our system allows for the quantitative analysis of key relative structural features between collagen fibers and cancer cell clusters. In addition to SHG, our multiphoton system is capable of imaging other endogenous fluorophores such as nicotinamide adenine dinucleotide (NADH) or flavin adenine dinucleotide (FAD) as well as any of the routine exogenous multiphoton probes used to stain tissue.

CONCLUSION

We present an imaging and analysis protocol that uses high content imaging techniques coupled with supervised learning to perform semi-automated TACS-3 scoring of slide mounted biopsy samples. We apply our technique to a previously annotated TMA containing tissue from 207 patients with invasive breast cancer. The resulting scores are shown to positively correlate with manual annotations and to predict patient outcome with good statistical significance. Future work will attempt to validate this technique on larger cohorts of breast cancer patients, to study ECM targeted drug responses in animal models, and to study collagen alignment in other cancers. As well, future work will focus on improving the clinical application of these techniques so they can be run by untrained clinical personnel and be run at the time of acquisition to find ROIs and TACS within those regions automatically. Together with more automation, TACS screening has great potential as a clinical diagnostic tool that can provide relevant prognostic information from large numbers of tissue samples.

ACKNOWLEDGMENTS

We would like to acknowledge assistance and valuable discussions with members of the Keely, Medical Devices and LOCI groups. We also acknowledge funding from NIH R01 Grants CA114462 and CA136590 to P.J.K and K.W.E, T32 CA009206 to J.S.B., and the Morgridge Institute for Research. The authors are grateful to Johannes Schindelin, Curtis Rueden, Andreas Velten, Ilya Goldberg, and Paolo Provenzano for their helpful technical suggestions and Andreas Friedl for providing access to the TMA. As well we wish to acknowledge the help of the WiscScan programmers at LOCI, Ajeet Vivekanandan, David Mayer and Mohit Chainani for their assistance in adding support for new functionality in WiscScan and debugging current features.

REFERENCES

1. Allred DC, Harvey JM, Berardo M, Clark GM. Prognostic and predictive factors in breast cancer by immunohistochemical analysis. Mod Pathol 1998;11:155-68.
2. Press M, Slamon D, Cobleigh M, Vogel C, Zhou JY, Anderson S, *et al.* Improved clinical outcomes for herceptin (R)-treated patients selected by fluorescence *in situ* hybridization (FISH). Mod Pathol 2002;15:47A.
3. Habel LA, Shak S, Jacobs MK, Capra A, Alexander C, Pho M, *et al.* A population-based study of tumor gene expression and risk of breast cancer death among lymph node-negative patients. Breast Cancer Res 2006;8:R25.
4. Nahta R, Esteva FJ. HER2 therapy: Molecular mechanisms of trastuzumab resistance. Breast Cancer Res 2006;8:215.
5. Shamir L, Delaney JD, Orlov N, Eckley DM, Goldberg IG. Pattern recognition software and techniques for biological image analysis. PLoS Comput Biol 2010;6:e1000974.
6. Madabhushi A, Agner S, Basavanhally A, Doyle S, Lee G. Computer-aided prognosis: Predicting patient and disease outcome via quantitative fusion of multi-scale, multi-modal data. Comput Med Imaging Graph 2011;35:506-14.
7. Myers G. Why bioimage informatics matters. Nat Methods 2012;9:659-60.
8. Rønnov-Jessen L, Petersen OW, Koteliansky VE, Bissell MJ. The origin of the myofibroblasts in breast cancer. Recapitulation of tumor environment in culture unravels diversity and implicates converted fibroblasts and recruited smooth muscle cells. J Clin Invest 1995;95:859-73.

9. Elenbaas B, Spirio L, Koerner F, Fleming MD, Zimonjic DB, Donaher JL, *et al.* Human breast cancer cells generated by oncogenic transformation of primary mammary epithelial cells. Genes Dev 2001;15:50-65.
10. Tlsty TD, Hein PW. Know thy neighbor: Stromal cells can contribute oncogenic signals. Curr Opin Genet Dev 2001;11:54-9.
11. Boyd NF, Martin LJ, Yaffe MJ, Minkin S. Mammographic density and breast cancer risk: Current understanding and future prospects. Breast Cancer Res 2011;13:223.
12. Guo YP, Martin LJ, Hanna W, Banerjee D, Miller N, Fishell E, *et al.* Growth factors and stromal matrix proteins associated with mammographic densities. Cancer Epidemiol Biomarkers Prev 2001;10:243-8.
13. Boyd NF, Martin LJ, Sun L, Guo H, Chiarelli A, Hislop G, *et al*. Body size, mammographic density, and breast cancer risk. Cancer Epidemiol Biomarkers Prev 2006;15:2086-92.
14. Boyd NF, Guo H, Martin LJ, Sun L, Stone J, Fishell E, *et al*. Mammographic density and the risk and detection of breast cancer. N Engl J Med 2007;356:227-36.
15. Boyd NF, Martin LJ, Bronskill M, Yaffe MJ, Duric N, Minkin S. Breast tissue composition and susceptibility to breast cancer. J Natl Cancer Inst 2010;102:1224-37.
16. Thurfjell E. Breast density and the risk of breast cancer. N Engl J Med 2002;347:866.
17. Habel LA, Dignam JJ, Land SR, Salane M, Capra AM, Julian TB. Mammographic density and breast cancer after ductal carcinoma *in situ*. J Natl Cancer Inst 2004;96:1467-72.

18. Boyd NF, Rommens JM, Vogt K, Lee V, Hopper JL, Yaffe MJ, *et al.* Mammographic breast density as an intermediate phenotype for breast cancer. Lancet Oncol 2005;6:798-80.

19. McCormack VA, dos Santos Silva I. Breast density and parenchymal patterns as markers of breast cancer risk: A meta-analysis. Cancer Epidemiol Biomarkers Prev 2006;15:1159-69.

20. Cil T, Fishell E, Hanna W, Sun P, Rawlinson E, Narod SA, *et al.* Mammographic density and the risk of breast cancer recurrence after breast-conserving surgery. Cancer 2009;115:5780-7.

21. Provenzano PP, Eliceiri KW, Campbell JM, Inman DR, White JG, Keely PJ. Collagen reorganization at the tumor-stromal interface facilitates local invasion. BMC Med 2006;4:38.

22. Provenzano PP, Eliceiri KW, Yan L, Ada-Nguema A, Conklin MW, Inman DR, *et al.* Nonlinear optical imaging of cellular processes in breast cancer. Microsc Microanal 2008;14:532-48.

23. Provenzano PP, Inman DR, Eliceiri KW, Knittel JG, Yan L, Rueden CT, *et al.* Collagen density promotes mammary tumor initiation and progression. BMC Med 2008;6:11.

24. Conklin MW, Eickhoff JC, Riching KM, Pehlke CA, Eliceiri KW, Provenzano PP, *et al.* Aligned collagen is a prognostic signature for survival in human breast carcinoma. Am J Pathol 2011;178:1221-32.

25. Provenzano PP, Inman DR, Eliceiri KW, Trier SM, Keely PJ. Contact guidance mediated three-dimensional cell migration is regulated by Rho/ROCK-dependent matrix reorganization. Biophys J 2008;95:5374-84.

26. Zipfel WR, Williams RM, Christie R, Nikitin AY, Hyman BT, Webb WW. Live tissue intrinsic emission microscopy using multiphoton-excited native fluorescence and second harmonic generation. Proc Natl Acad Sci U S A 2003;100:7075-80.

27. Zipfel WR, Williams RM, Webb WW. Nonlinear magic: Multiphoton microscopy in the biosciences. Nat Biotechnol 2003;21:1369-77.

28. Williams RM, Zipfel WR, Webb WW. Interpreting second-harmonic generation images of collagen I fibrils. Biophys J 2005;88:1377-86.

29. Chen X, Nadiarynkh O, Plotnikov S, Campagnola PJ. Second harmonic generation microscopy for quantitative analysis of collagen fibrillar structure. Nat Protoc 2012;7:654-69.

30. Burke K, Tang P, Brown E. Second harmonic generation reveals matrix alterations during breast tumor progression. J Biomed Opt 2013;18:31106.

31. Kakkad SM, Solaiyappan M, Argani P, Sukumar S, Jacobs LK, Leibfritz D, *et al.* Collagen I fiber density increases in lymph node positive breast cancers: Pilot study. J Biomed Opt 2012;17:116017.

32. Ajeti V, Nadiarnykh O, Ponik SM, Keely PJ, Eliceiri KW, Campagnola PJ. Structural changes in mixed Col I/Col V collagen gels probed by SHG microscopy: Implications for probing stromal alterations in human breast cancer. Biomed Opt Express 2011;2:2307-16.

33. Ambekar R, Lau TY, Walsh M, Bhargava R, Toussaint KC Jr. Quantifying collagen structure in breast biopsies using second-harmonic generation imaging. Biomed Opt Express 2012;3:2021-35.

34. Altendorf H, Decencière E, Jeulin D, De sa Peixoto P, Deniset-Besseau A, Angelini E, *et al.* Imaging and 3D morphological analysis of collagen fibrils. J Microsc 2012;247:161-75.

35. Pehlke C, Bredfeldt JS, Doot J, Sung KE, Provenzano P, Riching K*, et al.* Quantification of collagen architecture using the curvelet transform. Integrative Biology, in Review; January 2014.

36. Bredfeldt JS, Liu Y, Pehlke CA, Conklin MW, Szulczewski JM, Inman DR, *et al.* Computational segmentation of collagen fibers from second-harmonic generation images of breast cancer. J Biomed Opt 2014;19:16007.

37. Falzon G, Pearson S, Murison R. Analysis of collagen fibre shape changes in breast cancer. Phys Med Biol 2008;53:6641-52.

38. Rubbens MP, Driessen-Mol A, Boerboom RA, Koppert MM, van Assen HC, TerHaar Romeny BM, *et al.* Quantification of the temporal evolution of collagen orientation in mechanically conditioned engineered cardiovascular tissues. Ann Biomed Eng 2009;37:1263-72.

39. Bayan C, Levitt JM, Miller E, Kaplan D, Georgakoudi I. Fully automated, quantitative, noninvasive assessment of collagen fiber content and organization in thick collagen gels. J Appl Phys 2009;105:102042.

40. Baba F, Swartz K, van Buren R, Eickhoff J, Zhang Y, Wolberg W, *et al*. Syndecan-1 and syndecan-4 are overexpressed in an estrogen receptor-negative, highly proliferative breast carcinoma subtype. Breast Cancer Res Treat 2006;98:91-8.

41. Eliceiri K, Nazir M. Wiscscan, 2012. Available: http://www.loci.wisc.edu/software/wiscscan.[2012 Apr 04].

42. Linkert M, Rueden CT, Allan C, Burel JM, Moore W, Patterson A, *et al.* Metadata matters: Access to image data in the real world. J Cell Biol 2010;189:777-82.

43. Preibisch S, Saalfeld S, Tomancak P. Globally optimal stitching of tiled 3D microscopic image acquisitions. Bioinformatics 2009;25:1463-5.

44. Candes E, Demanet L, Donoho D, Ying LX. Fast discrete curvelet transforms. Multiscale Model Simul 2006;5:861-99.

45. Stein AM, Vader DA, Jawerth LM, Weitz DA, Sander LM. An algorithm for extracting the network geometry of three-dimensional collagen gels. J Microsc 2008;232:463-75.

46. Schindelin J, Arganda-Carreras I, Frise E, Kaynig V, Longair M, Pietzsch T, *et al.* Fiji: An open-source platform for biological-image analysis. Nat Methods 2012;9:676-82.

47. Ignacio AC, Kaynig V, Schindelin J. Trainable Weka Segmentation. Available: http://www.fiji.sc/Trainable_Weka_Segmentation.[2013 Oct 25].

48. Breiman L. Random forests. Mach Learn 2001;45:5-32.

49. Criminisi A, Shotton J, Konukoglu E. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. Found Trends Comput Graph Vision 2011;7:81-227.

50. Bresenham JE. Algorithm for computer control of a digital plotter. Ibm Syst J 1965;4:25-30.

51. Francis K, Palsson BO. Effective intercellular communication distances are determined by the relative time constants for cyto/chemokine secretion and diffusion. Proc Natl Acad Sci U S A 1997;94:12258-62.

52. Cox DR. Regression models and life-tables. J R Stat Soc Series B Stat Methodol 1972;34:187.

53. Nadiarnykh O, LaComb RB, Brewer MA, Campagnola PJ. Alterations of the extracellular matrix in ovarian cancer studied by Second Harmonic Generation imaging microscopy. BMC Cancer 2010;10:94.

54. Drifka CR, Eliceiri KW, Weber SM, Kao WJ. A bioengineered heterotypic stroma-cancer microenvironment model to study pancreatic ductal adenocarcinoma. Lab Chip 2013;13:3965-75.

55. Hu W, Zhao G, Wang C, Zhang J, Fu L. Nonlinear optical microscopy for histology of fresh normal and cancerous pancreatic tissues. PLoS One 2012;7:e37962.

56. Cox G, Kable E, Jones A, Fraser I, Manconi F, Gorrell MD 3-dimensional imaging of collagen using second harmonic generation. J Struct Biol 2003;141:53-62.

57. Lacomb R, Nadiarnykh O, Townsend SS, Campagnola PJ. Phase Matching considerations in Second Harmonic Generation from tissues: Effects on emission directionality, conversion efficiency and observed morphology. Opt Commun 2008;281:1823-32.

58. Stoller P, Kim BM, Rubenchik AM, Reiser KM, Da Silva LB. Polarization-dependent optical second-harmonic imaging of a rat-tail tendon. J Biomed Opt 2002;7:205-14.

59. Kliger DS, Lewis JW, Randall CE. Polarized Light in Optics and Spectroscopy. New York: Adademic Press; 1990.

60. Oldenbourg R. Polarization Microscopy with the LC-PolScope. Cold Spring Harbor, New York: Cold Spring Harbor Laboratory Press; 2005.

61. Junqueira LC, Bignolas G, Brentani RR. Picrosirius staining plus polarization microscopy, a specific method for collagen detection in tissue sections. Histochem J 1979;11:447-55.

62. Whittaker P, Kloner RA, Boughner DR, Pickering JG. Quantitative assessment of myocardial collagen with picrosirius red staining and circularly polarized light. Basic Res Cardiol 1994;89:397-410.

63. Rueden CT, Conklin MW, Provenzano PP, Keely PJ, Eliceiri KW. Nonlinear optical microscopy and computational analysis of intrinsic signatures in breast cancer. Conf Proc IEEE Eng Med Biol Soc 2009;2009:4077-80.

64. Conklin MW, Provenzano PP, Eliceiri KW, Sullivan R, Keely PJ. Fluorescence lifetime imaging of endogenous fluorophores in histopathology sections reveals differences between normal and tumor epithelium in carcinoma *in situ* of the breast. Cell Biochem Biophys 2009;53:145-57.

65. Wang W, Wyckoff JB, Goswami S, Wang Y, Sidani M, Segall JE, *et al.* Coordinated regulation of pathways for enhanced cell motility and chemotaxis is conserved in rat and mouse mammary tumors. Cancer Res 2007;67:3505-11.

66. Maller O, Hansen KC, Lyons TR, Acerbi I, Weaver VM, Prekeris R, *et al.* Collagen architecture in pregnancy-induced protection from breast cancer. J Cell Sci 2013;126:4108-10.

Tables

Table 1: Features used in epithelial cell segmentation and TACS-3 fiber classification tasks

Task	**Feature description**	**Total number of features**
Epithelial cell classification	Gaussian blur	5
	Sobel filter	5
	Hessian	40

	Difference of Gaussians	24
	Membrane projections	6
TACS-3 fiber classification	Fiber curvature	1
	Fiber width	1
	Fiber length	2
	Fiber density	9
	Fiber alignment	9
	Epithelial proximity	3
	Relative epithelial angle	2

TACS: Tumor associated collagen signatures

Table 2: Correlation between manual and supervised learning approaches

Scoring Method	**Correlation coefficient**	***P* value**
Manual score 1	0.295	2.7E-5
Manual score 2	0.311	0.9E-5
Manual score 3	0.271	1.2E-5

There is statistically significant positive correlation between the supervised learning approach and all manual scoring approaches

Table 3: Univariate Cox proportional hazard analysis results for various feature combinations

Feature Type	**Hazard ratio (DFS)**	***P* value (DFS)**	**Hazard ratio (DSS)**	***P* value (DSS)**

Fiber curvature	1.432	0.179	1.657	0.077
Fiber density	2.195	0.003	1.831	0.032
Fiber alignment	1.958	0.011	1.588	0.100
TACS-3 score	2.588	0.002	2.250	0.008

The TACS-3 scoring method that includes both fiber and cellular information produces the more significantly prognostic scores compared to fiber information alone. TACS: Tumor associated collagen signatures, DFS: Disease free survival, DSS: Disease specific survival

Table 4: Feature ranking based on SVM feature weight w for a 16 patient (8 TACS-3 positive and 8 TACS-3 negative) training set including the average feature value difference between the positive and negative training cases f_p - f_n

Feature description	w	f_p - f_n
Standard nearest align	0.716	−0.004
Nearest distance to bound	0.491	−0.086
Inside epithelial region	0.393	0.572
Standard nearest distance	0.348	0.211
Mean nearest distance	0.238	0.188
Curvature	0.219	0.010
Box density 128	0.190	−0.311
Width	0.183	−0.050
Box alignment 64	0.155	0.136

Box alignment 32	0.150	0.079
Box alignment 128	0.111	0.145
Box density 64	0.109	−0.249
Nearest relative boundary angle	0.087	0.004
Total length	0.079	0.004

SVM: Support vector machine, TACS: Tumor associated collagen signatures. These values show the trend of the feature values between TACS-3 positive and TACS-3 negative

Figure Legends

Figure 1: Block diagram of the tumor associated collagen signatures scoring system. The red steps are performed with WiscScan, the blue steps are performed with ImageJ/FIJI, and the green steps are performed with MATLAB based tools. The left side shows the steps performed on the second harmonic generation images, while the right side shows the steps performed on the bright field images. The middle column combines information from both modalities

Figure 2: Photo of imaging system (a) and optical block diagram panel (b). PMT = Photomultiplier tube, QCam = QImaging RGB camera, LED = Bright field lamp, DBS = Dichroic beamsplitter, TL = Tube lens, CL = Condenser lens, L = Lens, M = Mirror, BE = Beam expander, Z = Z-direction translation, XY = xy translation, XYG = xy galvonometer driven mirrors, PC = Pockel's cell, RAMM = Rapid Automated Modular Microscope (ASI), λ2, λ4 = half and quarter waveplates

Figure 3: Integrated fiber angle and epithelial boundary feature algorithms. Panel A shows a single fiber (green) and epithelial region boundary (red) with one highlighted fiber endpoint $\boldsymbol{v}$.

Zoomed versions of panel A are shown in panels B, C and D where individual image pixels are represented as filled circles. The nearest distance from $\boldsymbol{v}$ to the boundary is indicated in panel B, intersection between the endpoint extension line and the boundary is shown in panel C and the quadratic curve fit to the boundary at the intersection point and tangent line are shown in panel D

Figure 4: The forward second harmonic generation image was overlaid upon the bright field image for an entire H&E stained tissue microarray slide. Scale bar = 25 μm

Figure 5: Examples of tumor associated collagen signatures (TACS-3) positive and negative regions within our cohort of breast cancer samples. Collagen fibers (yellow) registered and overlaid onto bright field images of an H&E stained tissues. These images illustrate the features of the TACS-3 signature (green arrows, top row), particularly, straightened aligned collagen fibers that terminate at steep angles relative to epithelial region boundaries (a-c). Three TACS-3 negative cases are shown for comparison (red arrows, bottom row), where collagen fibers are wavy (d), terminate at adipose tissue (e), and wrap around epithelium (f) scale bar = 25 μm

Figure 6: Sample fiber extraction results. The resulting image after curvelet denoising (a) shows likely fiber pixels in white and likely background pixels in grey. The extracted fiber network is overlaid on the original second harmonic generation image (b) with many appropriate segmentations and a few under-and over-segmentations (green arrows). Scale bar = 50 μm

Figure 7: Sample epithelial cell segmentation results. The raw probability map produced by the trainable Weka Segmentation ImageJ plugin (a) is filtered to estimate epithelial cluster density (b) and thresholded (c) to produce epithelial region boundaries which are overlaid onto original bright field images to validate the segmentation (d) scale bar = 100 μm

Figure 8: KM curves for disease free survival and disease specific survival showing the prognostic classification produced by our supervised learning based tumor associated collagen signatures-3 scoring approach

Chapter 5: ImageJ2: ImageJ for the next generation of scientific image data

Adapted from: ImageJ: ImageJ for the next generation of scientific image data. Curtis T. Rueden, Johannes Schindelin, Barry E. DeZonia and Kevin W. Eliceiri. BMC Bioinformatics.

Background. The image analysis program ImageJ is a widely used tool for image analysis in the biological sciences and beyond. Due to its ease of use, flexible macro language and plug-in architecture, ImageJ has found itself being used effectively by the non-programmer, the amateur programmer, and the professional programmer alike. This is a feat that arguably no other imaging analysis package, commercial or Open Source, has achieved. ImageJ benefits from the contributions of many, including amateur programmers writing code for their own use, lab-hired professional programmers writing code for their clients, and the bench scientist who may not develop code but gives feedback and helps find new applications. This diversity of contributors has resulted in a unique community that spans the biosciences and physical sciences and has made ImageJ an invaluable resource. However, with the rapid growth in the ImageJ user base and broad range of applications, coupled with equally rapid advances in biological imaging technology, there was clear need for a concerted software engineering effort for ImageJ that would address current and future needs and yet still be the same ImageJ that this diverse user community all uses.

Results. We have implemented key infrastructural improvements with professional software engineering practices in ImageJ. This new next-generation ImageJ is fully backwards compatible while providing new functionality. Much of the development focus has been on core architecture, with a separation of the data model from the user interface allowing ImageJ to be run headlessly by other applications for improved interoperability, improvements to the plugin and scripting frameworks, and a broadening of the ImageJ data model to support other data types including large N-dimensional datasets.

Conclusions. Scientific imaging benefits from open source image analysis programs that advance new method development and deployment to a diverse audience. ImageJ has excelled in this regard, but due to new challenges is at a development crossroads that must be addressed. The described improvements in ImageJ attempt to address current and future needs leading to continued success and new innovation. Future efforts will focus on application development that takes advantage of this additional functionality.

Keywords

ImageJ, image processing, N-dimensional, interoperability, open source, open development process, microscopy software, light microscopy

Background

ImageJ [1] is a powerful, oft-referenced platform for image processing, particularly scientific image analysis in the life sciences, developed by Wayne Rasband at the National Institutes of Health (NIH). Since its initial release in 1997, it has been enormously useful in many scientific endeavors and projects. For sixteen years, the program has evolved and developed far beyond its originally intended scope. After such an extended period of sustained growth, any successful software project benefits from a subsequent period of scrutiny and refactoring, and ImageJ is no exception. Such review helps the program to remain accessible to newcomers, powerful enough for experts, and relevant to an evolving community. As such, we are developing a new version of ImageJ nicknamed ImageJ2, to build on its successful qualities while improving the core architecture of ImageJ to encompass the scientific needs of the next decade and beyond. Key motivations for the development of ImageJ2 include:

1) To support the next generation of image data. Over time, the infrastructure of image acquisition has grown in sophistication and complexity. In the field of microscopy, for example, we were once limited to study of single image planes, but with modern imaging techniques we can now record much more information: physical location in time and space (X, Y, Z, time), lifetime histograms across a range of spectral emission channels, polarization state of light, phase and frequency, angles of rotation (e.g., in selective plane illumination microscopy), and high-throughput screens including multiple plates and wells, just to name a few. The ImageJ infrastructure needs to be generalized and improved to ease working with these new paradigms of image data and beyond.

2) To interoperate & collaborate with other projects. As the Internet has developed and evolved, the field of software engineering has seen corresponding evolution and improvements to the available software development tools and infrastructure. In 2013, there are many examples of completely open, scalable software projects with geographically distributed contributors all over the world [2]. ImageJ is a perfect candidate for such a development model, since it is permissively open source, widely used by scientists and others, and highly extensible. While ImageJ1 has open source code and a public web site, ImageJ2 strives to foster maximum interoperability and collaboration.

3) To broaden the ImageJ community. ImageJ has the potential to be a powerful tool across a wide range of scientific disciplines, including life sciences, earth sciences, astronomy, fluid dynamics, computer vision... any field that benefits from image visualization, processing and analysis. By providing central online resources for ImageJ, from which users can not only benefit but also easily contribute to, we enhance its impact and utility throughout the scientific community.

Historical Context

ImageJ2 began in late 2009 as a project at the Laboratory for Optical and Computational Instrumentation (LOCI) in collaboration with the Marine Biological Laboratory (MBL) at Woods Hole, and the Broad Institute, thanks to a Grand Opportunities grant funded by the NIH. Shortly after being funded, we initiated a flurry of communication with the ImageJ community via the ImageJ mailing list, to solicit ideas and use cases. We were keenly aware that the Fiji project ("Fiji Is Just ImageJ") [3] had been pursuing similar goals since late 2007, and that Fiji ultimately needed an improved ImageJ core to facilitate more flexible N-dimensional scientific image analysis, so we made plans with the Fiji developers to work together on a unified ImageJ for the entire community. Curtis Rueden, lead developer of LOCI, attended a Fiji hackathon at EMBL in March of 2010, where many of the initial aspects of the collaboration were discussed, including usage of the powerful ImgLib2 library [4] as the basis for ImageJ2's N-dimensional data model, as well as how backwards compatibility with the previous version of ImageJ could be achieved. Later, at the ImageJ Conference in October 2010, we presented our initial plans, directions and progress and gathered additional feedback, with the next few months pursuing those refined development directions.

In September of 2011, Johannes Schindelin, the co-creator and maintainer of Fiji, moved to LOCI to work on ImageJ2 and Fiji full time. By then the line between ImageJ2 and Fiji was quite clear: ImageJ2 is the new core of ImageJ for next-generation image data, whereas Fiji is a suite of extensions tailored toward image analysis in the life sciences. Other distributions with other extensions for other fields should be equally possible; to facilitate that, we began migrating the infrastructural (i.e., non-life-science-specific) parts of Fiji into the ImageJ2 core, including the cross-platform launcher and powerful updater mechanism, among others.

Meanwhile, also in September of 2011, LOCI launched the SCIFIO project ("SCientific Image File Input and Output") to improve and generalize Bio-Formats [5], a library for input and output (I/O) of life sciences image file formats. SCIFIO fits in perfectly with ImageJ2, since file I/O is a prerequisite to analyzing data. SCIFIO now forms the backbone of ImgLib2 and ImageJ2's image I/O support, making it much easier to extend ImageJ with support for additional file formats.

In December of 2011, we had another ImageJ/Fiji hackathon at the Max Planck Institute of Molecular Cell Biology and Genetics (MPI-CBG) in Dresden, Germany, with a focus on interoperability. As part of our effort to grow the ImageJ community, we targeted several other software packages with which interoperability would be valuable: KNIME [6], OMERO [7] and Icy [8]. (In addition, by then we had also already completed initial integration with Broad's CellProfiler [9] software.) Participants from those three projects attended the hackathon, and together we founded the SciJava effort [10], making a commitment to work toward a common software stack with shared standards such as regions of interest (ROIs), rather than every project reinventing its own wheels.

2012 saw the release of the first beta version of ImageJ2, as well as several subsequent versions. We also attended the ImageJ Conference again in October, to present the project's current progress and offer initial workshops showing how to write plugins using ImageJ2 and ImgLib2.

Design Goals

Compatible. There are a vast number of existing extensions—plugins, macros and scripts—for ImageJ, which have proven extremely useful to the user community. Any new version of ImageJ must continue to support these extensions as faithfully as possible while also enabling necessary new functionality. We have even gone so far as to model the new ImageJ user interface after the current one as much as possible, to make it easier for existing users to migrate to the new version.

N-dimensional. ImageJ2 aims to provide robust support for image data beyond space and time. A few examples include multispectral and hyperspectral images; fluorescence lifetime measured in the time or frequency domains; multi-angle data from acquisition modalities such as selective plane illumination microscopy (SPIM); multi-position data from High Content Screens; and experiments using polarized light. However, the design is by no means limited to those examples; by providing a truly N-dimensional data model, any newly emerging modalities will be expressible in the new infrastructure.

Headless. To work "headless" means the software can run remotely on a server, a cluster or a cloud without a graphical user interface. In computer science terminology, ImageJ2 strives to have a proper separation of concerns between data model and display thereof, enabling headless operation and much more. ImageJ1 was developed with "single computer, single user, single operation" in mind, but ImageJ2 aims to provide a modular, multi-layered set of functions, with each layer encapsulated and building on the lower layers.

Extensible. The quality that makes ImageJ most powerful—its greatest strength—is its extensibility. From its inception [1], ImageJ has had a mechanism by which users can develop their own plugins and macros to extend its capabilities. Sixteen years later, a huge plethora of such plugins and macros have been shared and published. It is paramount that ImageJ2 also be easily modifiable and extendible by its user community. We have put a lot of thought into ImageJ2's plugin framework to make all sorts of extensions possible, and common ones easy.

Interoperable. There is no silver bullet in image processing. No matter how powerful ImageJ becomes, and no matter how many extensions exist, there will always be other equally powerful and useful tools as well. Users benefit most when exchanging information between such tools is as easy as possible. One of ImageJ2's primary motivations is to enable usage of ImageJ code from within other applications, and to support open standards for data storage and exchange.

Updatable. Given the goal of extensibility above, it is vital for ImageJ to keep track of such extensions: it needs a built-in, configurable update mechanism to manage extensions and keep the software up-to-date. And this update mechanism must be scalable and distributed, such that software developers can publish their own extensions on their own websites, without needing to obtain permission from "powers that be."

Openly developed. In the interest of transparency and reproducibility—especially in the context of open science—the ImageJ2 project strives to be as accessible as possible to everyone. An openly developed project is not only open source, but has an open process, including open history, open discussion forums, open issue tracking system, open roadmap, and an open patch submission process. Perfecting these qualities allows a project to be truly driven by the community, and greatly magnifies its relevance and utility worldwide.

Other Design Considerations

Minimize complexity. The complexity of a codebase is more than the number of lines of code. But each line of code does add complexity, making the project harder to understand and maintain. In simple terms, each line of code in a project is not an asset, but a liability. Code is fragile; any change to a program's code can have rippling consequences throughout every aspect of the program. Good program design minimizes such ripples via a modular, encapsulated design (see "Headless" above), and application of the "Don't Repeat Yourself (DRY)" concept. The less code, the easier it is to change and improve that code. Vital design considerations include modular code structure, careful management of dependencies, effective unit testing, and avoiding "clever" code which is needlessly difficult to understand. By keeping a clean, well-organized codebase of modest size, we facilitate contributions from the entire ImageJ community, as well as make maintenance of ImageJ feasible far into the future.

The ImageJ community includes both end users—who use ImageJ as an application and want it to "just work"—and software developers—who want to customize and invoke parts of ImageJ as a software library from their own programs. Many people think of "users" and "developers" as two different animals. But in actuality, the community spans a broad spectrum of people interested in both the user and developer paradigms: many users write simple scripts and macros to facilitate their image analysis, and many developers also use ImageJ as an application. With guidance, anyone can learn to be a developer, and one of the great things about ImageJ is that it makes it easy to get started. With ImageJ2, we want to make it even easier; for example, the Script Editor has migrated from Fiji to ImageJ2, and includes many tools to help people bridge that gap. It is hard enough just learning how to program a simple macro or script, much less dealing with compilers, IDEs and the command line.

The guiding idea behind complexity minimization is that of sensible defaults. The goal is to make simple things easy, and complicated things possible, by structuring the software in layers. The lowest layers expose the program's full power and complexity, offering many functions with many input parameters. Each layer up chooses common defaults for some of those parameters, reducing the number of functions and parameters, with the highest levels aiming for simplicity on par with the "big green Xerox button" which performs the most commonly desired task with no fuss. ImageJ2 embraces "convention over configuration" and use of standards: for example, ImageJ2 uses Maven [11] for its build system to minimize the learning curve of working with the ImageJ source code.

Performance. Good performance in both time and space is one of ImageJ2's central goals. In some cases, such as with the ImageJ1 macro language, we actually hope to substantially improve time performance. And ImgLib2's flexible container model will allow us to often improve space performance in a transparent way, too. We have run benchmarks comparing the time performance of ImgLib2 with ImageJ1 as well as raw arrays; we found that ImgLib2 is not only highly comparable to ImageJ1 in that regard thanks to Java's Just-in-Time (JIT) compiler, but also enables processing of data structures not even possible within ImageJ1 [4].

That said, since ImageJ2 is still in beta, performance has not been our primary focus yet. We believe that compromising early for the sake of performance often results in a long-term increase in complexity. There is no doubt that performance must be a design consideration, but we believe that aggressive performance optimization should only be done on an as-needed basis, once the software is mature. Our approach is to think about performance enough to avoid limiting ourselves to a poorly-performing design, but not so much that the design becomes compromised for the sake of incremental performance gains.

Collaboration and open development. Ultimately, we want to spur the community to improve ImageJ in a collaborative way, by providing open access to resources. Of course, we recognize the need for responsive, reliable maintainers to serve as coordinators and facilitate contributions. But with the Internet's modern software infrastructure, it is now quite feasible to push ImageJ development in a more collaborative and community-driven direction, embracing the "GitHub effect" [12] of worldwide, distributed development.

ImageJ2 is not only open source, but follows an open development process, which we believe is an optimal fit for open scientific inquiry [13]. The project is open source with an open revision history [14], has open mailing lists [15], an open issue tracking system [16], an open roadmap [17], and open patch submission process [18]. We want to minimize the entry barrier for the community to not only use ImageJ2, but to contribute to it. We discuss development issues on public, archived mailing lists so that interested parties—even people who come late to the discussion—can stay informed of, and affect, the project's directions. This transparency also facilitates sensible, defensible software development processes, and fosters accountability amongst those involved in the project.

Implementation

The term "ImageJ" refers to much more than just an end-user application for N-dimensional image processing. ImageJ is also: a reusable library for N-dimensional image processing; an environment for rapid development of new image analysis workflows; and an extensible framework enabling the addition of new functionality at every level of the system. The following sections describe ImageJ's main features.

Data model. ImageJ2 has an entirely new data model for images and beyond. While ImageJ1 is inherently limited to five dimensions (X, Y, Z, time and channel), ImageJ2 has an N-dimensional data model, meaning it supports any number of dimensional axes. There are several presets, or you can define your own axis types. When visualizing multi-channel data, each channel can have its own LUT, similar to ImageJ1's composite image feature. But unlike ImageJ1, which is limited to a maximum of seven composited channels, ImageJ2 has no limit to the number of channels which can be composited together.

The ImageJ2 data model is driven by the ImgLib2 library, which provides an interface-driven design that supports numeric and non-numeric data types (8-bit unsigned integer, 32-bit floating point, etc.) in an extensible way. It also provides great flexibility regarding the source and structure of the data. Out of the box, ImgLib2 provides several data sources and sample organizations, including using a single primitive array, one array per plane, N-dimensional array "cells" cached to and from disk on demand, and planes read on demand from disk. But the library is general enough that alternative structures are also feasible. To quote the ImgLib2 paper [4]:

> The core paradigm [of ImgLib2] is a clean separation of pixel algebra (how sample values are manipulated), data access (how sample coordinates are traversed) and data representation (how the samples are stored, laid out in memory or paged to disc). ImgLib2 relies on virtual access to both sample values and coordinates, facilitating parallelizability and extensibility.

In other words, ImgLib2 has the same ultimate goal as Java does: “write once, run anywhere.” With ImageJ1, it is often necessary to introduce extensive case logic to handle the various pixel types. In ImgLib2, you write the algorithm in a general way, and it will work the same regardless of the input data’s specific type and storage mechanism (see Figure 1, right).

[Figure 1: IJ1 case logic compared to a unified ImgLib2 implementation.]

Thanks to ImgLib2, ImageJ2 supports many more numeric types than ImageJ1 does (see Table 1, right). In addition to the types shown in Table 1, arbitrary precision integer and decimal types are planned, and further types are definable using ImageJ2's flexible plugin framework. ImageJ2 also takes advantage of ImgLib2's extensible container system, which enables data to be stored internally however you want: as files on disk, remote URLs, within a database, generated on-the-fly, etc. As such, you can write an ImageJ2 command once and it will work with data in any pixel type, and stored internally using any storage container. Such commands can even be used with pixel and storage types implemented well after the original command, without having to change the original implementation.

[Table 1: Numeric types supported by ImageJ.]

In the course of developing ImageJ2, in collaboration with the KNIME Image Processing developers at Universität Konstanz, we have also enhanced the ImgLib2 library to better support ImageJ2's requirements. ImgLib2 now has a functional subsystem known as OPS which provides a higher-level computational framework on top of ImgLib2's iteration-based approach. OPS offers transformation and composition operations applied across entire images, and is a natural fit for many of ImageJ's image processing commands, such as the Image Calculator.

Regions of interest. ImageJ2 makes a distinction between a *region of interest* (ROI)—which is a function that identifies samples upon which to operate—and an *overlay*, which is a visual superimposed over a dataset, often (but not always) linked to a ROI. ImageJ2 uses the ROI model of ImgLib2, but builds its own overlay logic on top for visualization purposes. ImageJ2 allows any number of simultaneous overlays.

Because ROIs are part of the core ImgLib2 library, it is possible to process subsets of images identified by one or more ROIs, using an ImgLib2-based algorithm—and the OPS library can process data within a ROI as a single functional operation. Similar to other aspects of ImgLib2 and ImageJ2, image processing algorithms need not add explicit case logic to handle ROIs, but instead simply provide a pixelwise function (in the case of OPS) or iterate using ImgLib2's generic iteration mechanism (in the case of other algorithms).

Lastly, we are collaborating with other members of the SciJava initiative, including the OMERO and CellProfiler teams, to define a common ROI specification that we can all use to ensure interoperability across our projects.

File formats. ImageJ2 and ImgLib2 use the SCIFIO library ("SCientific Image Format Input and Output") to read and write scientific image file formats. SCIFIO is a new version of Bio-Formats—a popular plugin for ImageJ1—which has been generalized beyond the life sciences, making it a perfect fit for ImageJ2, since it also targets a broader scientific field than only life sciences, Fiji's purview. SCIFIO and ImageJ2 share a common mechanism for extensibility (see "Service architecture and SciJava Common" below), meaning that adding support for a file format to SCIFIO is very similar to writing an ImageJ plugin, and automatically makes the format supported in ImageJ as well as many other software packages such as MATLAB, the Insight Toolkit (ITK) [19] and OMERO. Like ImageJ2, SCIFIO provides a thorough separation of concerns: detecting the format of a particular file is a separate operation from parsing a file's metadata or extracting any of its pixels, making SCIFIO a flexible Swiss Army knife reusable from other software.

Compatibility. While ImageJ2 is a complete rewrite of the ImageJ core with an entirely new data model, it provides backwards compatibility by including a copy of the latest version of ImageJ 1.x internally. ImageJ2 has a “legacy layer” which launches an invisible ImageJ1 in the background which handles execution of plugins, macros and scripts intended for use with ImageJ1. The legacy layer transparently translates between “legacy” and “modern” data structures, sharing memory (i.e., wrapping by reference) whenever possible, so that the results of ImageJ1-specific operations remain available in the ImageJ2 environment. This approach allows ImageJ1 and ImageJ2 to continue to develop in tandem, with new ImageJ1 features continuing to be available in ImageJ2 via the legacy layer.

Naturally, the legacy layer inherits the limitations of ImageJ1: a limited set of data types, limited number of dimensions, no headless operation, no concurrent macro execution, only one instance of ImageJ possible per Java Virtual Machine (JVM), etc. But these limitations affect only operations that are provided as ImageJ1 plugins or macros. The legacy layer uses an advanced Java technique known as bytecode manipulation to make ImageJ1 aware of the ImageJ2 facilities: callbacks are inserted into selected code locations of ImageJ1 using a component called Javassist [20]. These callbacks guarantee that the internal state of the ImageJ1 and ImageJ2 instances are kept in synchrony, despite the fact that ImageJ1 was not designed to support such synchronization. In essence, ImageJ2 “rewrites” portions of ImageJ1 at runtime to make the legacy integration possible.

The legacy layer includes full support for non-interactive ImageJ1 commands, and for ImageJ1's macro language. However, interactive commands such as “Brightness & Contrast” are not possible to support via the legacy layer, due to the fact that they continuously manipulate legacy data structures—keeping them continuously in sync with ImageJ2 would not be performant enough. To resolve this problem, we have rewritten many such interactive commands directly as ImageJ2 commands. This solution suffices in most cases, since it is our eventual goal to rewrite all core ImageJ1 commands as ImageJ2 commands, freeing users from the need to use a legacy layer at all.

Further, to provide ultimate backwards compatibility in all cases, we have implemented a “legacy mode”: you can switch ImageJ2 completely back to ImageJ1 mode, hiding the ImageJ2 user interface and showing the ImageJ1 UI instead, by choosing “Switch to Legacy Mode” from the Help menu. The program then behaves identically to how ImageJ1 did, because it *is* ImageJ1 for all intents and purposes. The only difference is that when you are finished performing legacy operations, you can switch back to ImageJ2 mode (Help > Switch to Modern Mode) and the current state will be resynchronized to ImageJ2, allowing you to continue your work with the full power of ImageJ2.

Lastly, we have recently added limited support for ImageJ2-specific syntax and data structures to ImageJ1 as part of Fiji, since internally, Fiji already ships ImageJ2 as a library. This support comes in two forms:

1. ImageJ1 plugins may utilize an ImageJ2 application context to perform operations only

possible with ImageJ2. For example, ImageJ1's support for spreadsheet-style tables of values (called "results tables" in ImageJ parlance) are limited to numerical values only, except for a single header column, whereas in ImageJ2 there is no such limitation: any cell can have any object such as a string. Thus, it is now easy to create a plugin that mostly uses the ImageJ1 API as usual, but ultimately generates an ImageJ2-specific "spreadsheet" results table by mixing in ImageJ2 calls (see Figure 2, top right).

2. ImageJ1 plugins may use ImageJ2's declarative command syntax (i.e., the @Parameter annotation) to populate input values with less boilerplate code. In some cases, the reduction in code can be 50% or more (see Figure 3, bottom right, for an example).

[Figure 2: A mixed-world ImageJ1 + ImageJ2 command.]

[Figure 3: Comparison of pure ImageJ1 command with one using ImageJ2's declarative syntax.]

Service architecture and SciJava Common. ImageJ2 is divided into modular pieces of functionality known as services. For example, the animation service provides methods for animating: starting, stopping, setting frames per second, and so on. Each service may depend on other services, creating a hierarchy. ImageJ2 is nothing more than a collection of these services. ImageJ has many core services built in: plugins, events, displays, UIs, overlays, tools, I/O, options, platforms and others. But just as importantly, you can also create extensions which define their own services or override the default behavior, making ImageJ2 truly extensible and ultimately configurable.

In the course of developing ImageJ2, we facilitated this design by creating a service architecture (also called an "application container"). These underpinnings proved to be so useful, and so general, that we decided to split it into its own library called SciJava Common, part of the SciJava umbrella. This core library now forms the basis of ImageJ2's extensibility mechanism, as well as that of SCIFIO, which had begun to reinvent its own application container as well. Since these two projects now share the same base application context, it improves interoperability between them, ensuring that developers who understand one system can easily understand the other, since the same standard principles apply.

[Technical sidebar: For those software engineers familiar with the ideas of dependency injection (DI), inversion of control (IoC) and application containers such as Spring and Plexus, SciJava Common is quite similar—just tailored for use within the SciJava software stack. In principle, it would be possible to migrate the SciJava infrastructure to another platform such as Spring, though of course a careful cost/benefit analysis would need to be performed before doing so.]

The SciJava Common library itself provides the most central services, including:

- A **plugin service**, which dynamically discovers available plugins using the SezPoz library [21]. The plugin index is used to bootstrap the application context, since services are themselves a type of plugin.
- An **event service**, which provides a hierarchical publish/subscribe model for event handling using the EventBus library [22].
- A **log service**, for environment-agnostic data logging.

- An **object service**, which keeps a central typed index of available objects.
- A **thread service**, which manages a thread pool and dispatch thread(s) for code execution.

The SezPoz and EventBus libraries are carefully and completely encapsulated, meaning that while they are transitive dependencies, downstream code never needs to reference any of their classes directly. This approach has the advantage that the underlying implementation of the plugin and event services can be completely replaced to eliminate those dependencies, should such a thing be needed by a downstream project. Similarly, the default log service implementation uses the standard error stream, but an alternate SLF4J-based log service implementation is also provided, with other logging implementations equally possible. The SciJava Common library uses an interface-driven design for all services, allowing their behavior to be fully overridden as needed.

Plugin framework. The ImageJ2 plugin framework, built on top of SciJava Common, is a modular and extensible infrastructure for adding features. Plugins can now take many forms, including image processing, new tools, and even completely new displays. It is even possible to define your own types of plugins.

In ImageJ1, there are three kinds of plugins: regular PlugIns, which provide a freeform `run(String arg)` method; PlugInFilters, which process images one plane at a time; and PlugInTools, which implement tools that define how the user’s mouse input translates into modifications to an image when that tool is selected on the toolbar. In ImageJ2, these ideas still exist in the form of Commands and Tools, which are two of many different plugin types available. A Command defines an operation with typed inputs and outputs, while a Tool maps user input (primarily keyboard and mouse activity) to behavior. But many other types of plugins exist, each one defining a unique extension point to ImageJ (see Table 2, right).

[Table 2: Core plugin types.]

ImageJ2 commands. Commands are the "bread and butter" sort of plugin in ImageJ2, defining an operation with typed inputs and outputs. These types can be any Java class or primitive. When ImageJ2 executes a command, it automatically takes care of determining the values of its inputs during a *preprocessing* step which can be completely customized. By default, certain types of inputs (such as services) are filled in automatically, and then the remaining inputs (such as numbers and strings) are populated by user input via a dialog box created by the user interface (UI). The end result is much more declarative than ImageJ1 plugins are, freeing software developers from the need to explicitly ask the user for input values in the vast majority of cases. Moreover, this mechanism makes commands truly independent of the user interface, allowing them to work with any UI or headless. The command simply declares its inputs and outputs using the @Parameter annotation, and lets ImageJ do the rest. This declarative approach often substantially reduces boilerplate and UI-specific code, making commands shorter and easier to understand (see Figure 3, above).

Interoperability with other software. There is no one-size-fits-all tool for scientific image processing. A diversity of tools is good for users—as long as they can interoperate with each other. Our primary goal is for ImageJ2 to be reusable from other software. Because the plugin framework is completely headless, several software packages already take advantage of ImageJ2, including CellProfiler, KNIME Image Processing (KNIP) [23] and Alida and Grappa [24]. Even tools written in languages besides Java (e.g., CellProfiler is written in Python) have successfully made ImageJ2 commands available within their own paradigms. Since CellProfiler, KNIME and Alida are all workflow-based, it is logical to expose each ImageJ2 command as a node/module, with a clean mapping of the command's inputs & outputs to node/module inputs & outputs. In this way, an image processing algorithm implemented as an ImageJ2 command is "written once and runs anywhere," and can be automatically discovered and shared across a plethora of other software.

Flexible user interface. Another important aspect of interoperability is the ability to create alternative user interfaces (UIs). ImageJ2 provides a UI layer independent of any specific framework such as Swing, enabling developers to create their own UI, display and widget plugins which provide alternative user interfaces. ImageJ2 is even capable of displaying multiple UIs simultaneously in the same Java runtime. Making ImageJ support multiple different user interfaces in this way has been a challenge, but we believe it will be well worth it for many years to come: one very common question on the ImageJ mailing list is how to customize ImageJ, and doing so is now much easier with ImageJ2. In particular, it is no longer necessary to modify the existing source code—only to write a new user interface plugin (and any needed corresponding display and widget plugins) for your needs. While the flagship user interface for ImageJ2 is written in Swing and modeled after ImageJ 1.x (see Figure 4, right), we have proof-of-concept implementations in other UI frameworks such as Eclipse's Standard Widget Toolkit (SWT), Java

Abstract Windowing Toolkit (AWT) sans Swing, and Apache Pivot. We have also been successful in "reskinning" the ImageJ Swing UI with various Java Look & Feels (L&Fs), including the Metal, Motif, Nimbus, Aqua, Windows and GTK L&Fs.

[Figure 4: ImageJ2 in action.]

Launcher. ImageJ2 features a cross-platform launcher, which was formerly part of Fiji. In addition to ImageJ2, the launcher supports launching Fiji and ImageJ1. By having a single code base for all platforms, it reconciles the different ways to run ImageJ into a unified, consistent method; e.g., by supporting the same set of command-line options, supporting custom settings in an *ImageJ.cfg* file not only on Windows but everywhere, etc. The launcher is capable of running batch scripts in a number of languages (ImageJ macro, BeanShell, Jython, etc.) from the command line. Similar to Fiji's headless mode with no user interface, the *--headless* command-line option will launch ImageJ in headless mode, and the launcher will be extended to support other startup modes of ImageJ2, too. The Java Runtime Environment is discovered in a consistent manner across platforms, a sensible amount of memory to be allocated to the Java virtual machine is determined upon startup, and native library directories are automatically registered. For older versions of Mac OS X where only Java versions older than 1.6 are available, the launcher activates a translator component called Retrotranslator [25] which compensates for it. The launcher is written in portable C, and hence can be maintained more easily than previous ImageJ launchers, and ImageJ2's build server publicly provides up-to-date versions for all prevalent platforms: Windows, Mac OS X and Linux, each 32-bit as well as 64-bit. The launcher also supports various customizations, such as attaching an icon to the executable on Windows or Linux.

Updater. Fiji's most useful contribution to the ImageJ community has been, without a doubt, its updater. In its initial form, the Fiji updater informed users automatically when a new plugin version was uploaded to the Fiji website, and enabled single-click upgrading to the new version. It was clear that such a useful feature belonged in the core of ImageJ2, so we took steps to migrate it there. In preparation, the updater was extended to follow multiple update sites, so that core ImageJ2 code could be served from a separate update site than Fiji-specific code. Consequently, the updater is fully backwards compatible with the Fiji update site, and Fiji now uses the ImageJ2 updater instead. By moving the updater to ImageJ2, not only is it visible to a wider audience than only Fiji users, it is also more robust as a consequence of automatic regression tests added for use with ImageJ2's build server.

Another advantage of supporting multiple update sites is that users can now set up their own update site to distribute their extensions, bringing the full power of automatic updates to those extensions, without them needing to become a part of the core ImageJ or Fiji distribution. This distributed model of update site fits in very well with the community-driven aspects of ImageJ. The update site approach is also much easier for most users than requiring them to download a JAR file and drop it into the plugins folder, although that still works with ImageJ2, too—but in that case, users will always have to update those plugins manually, and the developers will have to manually manage their plugin dependencies rather than the updater taking care of it automatically.

To make setting up third-party update sites even easier, we improved the updater to support an extensible upload mechanism, and provided support for several common upload methods including SSH, SFTP, WebDAV and the local filesystem. It is even possible to provide an update site via a free file syncing service such as Dropbox [26].

Several organizations have already set up their own update sites [27], and thanks to the updater together with the legacy layer, all of the plugins served from those sites are accessible within ImageJ2. This includes over 350 Fiji plugins, as well as dozens more from third-party update sites such as LOCI, the Stowers Institute, and the BioImaging and Optics platform of the EPFL.

Results and Discussion

The current ImageJ2 implementation is a modular, flexible, extensible open source application and library fully backwards-compatible with ImageJ1, with a software architecture that was redesigned from scratch to prepare it for current and future needs of image processing in life sciences. It is developed using an open process in close collaboration with related projects, with a strong emphasis on providing reliable, maintainable software.

Backwards compatibility is maintained by providing a legacy layer that can execute existing ImageJ1 macros and plugins (within ImageJ1 limitations, of course). To run legacy plugins that integrate tightly with ImageJ1's user interface (and hence cannot integrate with ImageJ2's UI), it is possible to switch to the legacy mode which displays the ImageJ1 UI and hides the ImageJ2 one, then switch back when finished.

Thanks to the improvements in the ImgLib2 library driven by the demands of next-generation imaging modalities, images of arbitrary size and number of dimensions can be processed, whereas ImageJ1 plugins are limited to five pre-defined dimensions and less than 2^{31} pixels per XY plane (e.g., 50,000 × 50,000 would be too large). Unless limited explicitly, ImageJ2 commands are completely agnostic of data type, number of dimensions, and even how the images are represented in memory. For example, if an algorithm such as a Gaussian convolution needs to be performed on an quaternion image, all one needs to do is to implement a quaternion type; the Gaussian convolution (and many other ImageJ2 commands) can be applied unchanged to any image using the newly-defined type, with no code changes to the specific algorithm itself. Commands no longer have to support virtual stacks explicitly because data access using ImgLib2 abstracts the implementation details of how images are represented away from the algorithms.

Extending ImageJ with new commands has been greatly simplified by the introduction of a declarative command paradigm. Rather than requiring the developer to invoke UI dialogs asking the user for input—an approach which can be very difficult to keep compatible with scripting and headless operation—ImageJ2 enables developers to declare an algorithm’s inputs and outputs directly. Such declarative commands are highly interoperable, which has been demonstrated by integrating support for ImageJ2 commands within CellProfiler and KNIME. Further efforts to foster interoperability even further are underway; in particular, we are working on improved integration between Java and native software so that ImageJ can interoperate with C++ libraries such as ITK.

As of this writing, ImageJ2 implements most of ImageJ1's functionality for use with unlimited data types, unlimited number of dimensions and arbitrary size. Existing ImageJ1 plugins which have not yet been ported to ImageJ2 can still be executed directly within ImageJ2 via the legacy layer. While certain types of legacy plugins do not behave well within ImageJ2's architecture, such as those integrated very tightly with ImageJ1's UI, such plugins are still accessible by switching to the legacy mode, which behaves exactly as ImageJ1 does. Our main focus thus far has been on robustness and correctness, so a large part of ImageJ2 has not yet been optimized for optimal time and memory performance; nonetheless, the current performance is comparable to ImageJ1 in many cases.

Some specific advantages of ImageJ2 over its predecessor include:

- ImageJ1 supports only five pixel types: 8-bit unsigned integer grayscale, 8-bit with a color lookup table, 16-bit unsigned integer grayscale, 32-bit floating point, and a 32-bit integer-packed color type representing three 8-bit unsigned color channels: red, green and blue. In contrast, ImageJ2 natively supports twelve different pixel types, with legacy support for others, and an extensibility layer to add new types as needed without needing to modify the ImageJ2 core when doing so. ImageJ1 sometimes requires case logic for algorithms to properly handle each of its supported pixel types; the ImgLib2 library provides a type-agnostic solution for algorithms to avoid repeating code, which helps to reduce the frequency of bugs.
- ImageJ1 can represent multiple color channels, z-slices and timepoints, in effect having

up to five-dimensional images (X, Y, Z, channel, time), but not more. It supports color compositing of samples over the channel dimension, but only if there are seven channels or fewer. Conversely, ImageJ2 supports any number of dimensional axes, with color compositing available for any number of channels. And thanks to ImgLib2's extensible data representation mechanism, ImageJ2 can also work with larger image planes than ImageJ1 can.

- ImageJ1 was designed to run by a single user on a single desktop computer. Many aspects of the program are structured as singletons: one macro interpreter, one WindowManager, one active image, one PlugInClassLoader, one active ROI per image, one set of overlays, one active ROI in the ROI manager, etc. This structure imposes many limitations; for example, multiple macros cannot run concurrently, and it is not possible to embed more than one instance of ImageJ1 on the same web page (since many web browsers allow only one JVM per web page). ImageJ2 is structured as an application container, avoiding the singleton pattern and hence many of ImageJ1's limitations. Multiple instances of ImageJ2 can run in the same JVM, each with multiple (or no) user interfaces, and each with multiple operations running concurrently.
- ImageJ1's user interface is written in Java AWT, with many assumptions throughout the codebase relying on this fact. Hence, ImageJ1 is only limitedly usable in a headless way, and it is not very feasible to embed it into another application using a different UI framework such as Swing or Eclipse SWT. ImageJ2 provides a clean separation of concerns between the underlying data model and the user interface, enabling it to run headless or within a variety of different user interface paradigms.
- ImageJ2 follows an open development process, which provides many benefits over the

centralized process of ImageJ1. For example, ImageJ2 minimizes the barrier to community contributions via an open patch submission process. We also maintain a continuous integration (CI) server which runs a number of automated regression tests whenever modifications to ImageJ's source code are published. Such tests help to avoid and detect regression bugs, so that core functionality and behavior will continue to work reliably as the program evolves.

Our next milestone for ImageJ2 will be to finalize the core architecture and come out of beta, releasing the final 2.0.0 version of the software. From that point forward, the focus will become more about extending ImageJ2's capabilities toward specific applications, fully leveraging the new system. Of course, we will continue adding features and improving usability; e.g., we will improve the updater to make it more user friendly, with a simpler and more intuitive user interface. We also plan to improve the documentation and tutorials, as well as creating new online resources such as a central directory of available plugins, aggregated from multiple update sites across the world. Lastly, we will continue to pursue additional avenues of interoperability, such as integration with the OMERO server, implementing the ability to invoke ImageJ2 commands headlessly from the server. The complete ImageJ2 roadmap, including a detailed breakdown of specific issues, can be found on the ImageJ2 web site.

Conclusions

Over the last fifteen years, ImageJ has grown organically as requested features have been added, with many contributions from outside developers. The result has been a program with a wide range of functionality capable of solving a diverse collection of image processing and analysis problems, particularly in the life sciences. However, this pattern of growth, even when carefully managed, is no substitute for a holistically engineered package built from the ground up with a modular design. Any successful software project, after a period of sustained growth and the addition of functionality outside the scope of the program's original intent, will benefit from an extended period of examination and refactoring, and ImageJ is no exception. Based on the pressing unmet needs of the existing ImageJ community and the needs of several open source projects that have met with difficulties attempting to build on ImageJ due to its limitations, we have over the last several years been designing and implementing a new ImageJ (nicknamed "ImageJ2") that employs best practices and proven software engineering approaches. ImageJ2 directly addresses two major needs: supporting applications where the underlying ImageJ data engine wasn't sufficient, and those where the lack of an underlying robust software design was limiting new functionality. The challenge has been to implement these practices and "harden" the ImageJ application without alienating or fragmenting its widespread user base that relies on it as an everyday research tool. As such, much of the development of ImageJ2 to date has stressed not only innovation but also compatibility with legacy code.

Both the new system and support for legacy plugins are now in place, allowing future work on ImageJ to focus less on architecture development and more on applications that take advantage of the new ImageJ's underlying engine and structure. These improvements to ImageJ not only allow it to be a powerful and flexible image processing and analysis application in its own right, but also a framework for interoperability between a plethora of external image visualization and analysis programs. In summary, no other scientific image processing program: 1) is open-source, cross-platform and public domain, enabling widespread adoption, while also 2) having such a huge collection of existing functionality, and 3) having a correspondingly large and diverse community in place to continue to reap the benefits as improvements are made.

Availability and requirements

Project name: ImageJ
Project home page: http://imagej.net/
Operating system(s): Platform independent
Programming language: Java
Other requirements: Java 1.6 or higher
License: Simplified (2-clause) BSD
Any restrictions to use by non-academics: None

List of abbreviations used

Alida: Advanced Library for Integrated Development of data analysis Applications
AWT: Abstract Windowing Toolkit
BSD: Berkeley Software Distribution
CI: Continuous Integration
DI: Dependency Injection
EMBL: European Molecular Biology Laboratory
EPFL: École Polytechnique Fédérale de Lausanne
Fiji: Fiji Is Just ImageJ
GIMP: GNU Image Manipulation Program
GNU: GNU's Not Unix
GTK: GIMP ToolKit
I/O: Input/Output
ImageJ1: ImageJ version 1.x, the legacy version of ImageJ
ImageJ2: ImageJ version 2.x, the new version of ImageJ
IoC: Inversion of Control

ITK: Insight ToolKit
JAR: Java ARchive
JIT: Just-In-Time compiler
JVM: Java Virtual Machine
KNIME: KoNstanz Information MinEr
KNIP: KNIME Image Processing nodes
L&F: Look & Feel
LOCI: Laboratory for Optical and Computational Instrumentation
MBL: Marine Biological Laboratory
MPI-CBG: Max Planck Institute of Molecular Cell Biology and Genetics
OME: Open Microscopy Environment
OMERO: OME Remote Objects
OS: Operating System
ROI: Region Of Interest
SCIFIO: SCientific Image File Input and Output
SFTP: Secure File Transfer Protocol
SLF4J: Simple Logging Facade for Java
SPIM: Selective Plane Illumination Microscopy
SSH: Secure SHell
SWT: Standard Widget Toolkit
UI: User Interface
URL: Uniform Resource Locator
WebDAV: Web Distributed Authoring and Versioning

References

[1] Schneider CA, Rasband WS, Eliceiri KW: **NIH Image to ImageJ: 25 years of image analysis.** *Nature Methods* 2012, **9(7):**671-675.
http://scholar.google.com/scholar?cluster=8490320759898335020&hl=en&as_sdt=0,50

[2] Feller J, Fitzgerald B: *Understanding Open Source Software Development*. London: Addison-Wesley; 2002.
http://scholar.google.com/scholar?cluster=9954603612042260275&hl=en&as_sdt=0,50

[3] Schindelin J, Arganda-Carreras I, Frise E, Kaynig V, Longair M, Pietzsch T, Preibisch S, Rueden C, Saalfeld S, Schmid B, Tinevez J, White DJ, Hartenstein V, Eliceiri K, Tomancak P, Cardona A: **Fiji: an open-source platform for biological-image analysis.** *Nature Methods*

2012, **9(7):**676-682.
http://scholar.google.com/scholar?cluster=17249863664147333646&hl=en&as_sdt=0,50

[4] Pietzsch T, Preibisch S, Tomančák P, Saalfeld S: **ImgLib2–Generic Image Processing in Java.** *Bioinformatics* 2012.
http://scholar.google.com/scholar?cluster=17114668151840635304&hl=en&as_sdt=0,50

[5] Linkert M, Rueden CT, Allan C, Burel J, Moore W, Patterson A, Loranger B, Moore J, Neves C, MacDonald D, Tarkowska A, Sticco C, Hill E, Rossner M, Eliceiri KW, Swedlow JR: **Metadata matters: access to image data in the real world.** *The Journal of Cell Biology* 2010, **189(5):**777-782.
http://scholar.google.com/scholar?cluster=17774693189715989612&hl=en&as_sdt=0,50

[6] Berthold MR, Cebron N, Dill F, Gabriel TR, Kötter T, Meinl T, Ohl P, Sieb C, Thiel K, Wiswedel B: **KNIME: The Konstanz information miner.** *Data Analysis, Machine Learning and Applications* 2008, 319-326.
http://scholar.google.com/scholar?cluster=16514357352138177203&hl=en&as_sdt=0,50

[7] Allan C, Burel J, Moore J, Blackburn C, Linkert M, Loynton S, MacDonald D, Moore WJ, Neves C, Patterson A, Porter M, Tarkowska A, Loranger B, Avondo J, Lagerstedt I, Lianas L, Leo S, Hands K, Hay RT, Patwardhan A, Best C, Kleywegt GJ, Zanetti G, Swedlow JR: **OMERO: flexible, model-driven data management for experimental biology.** *Nature Methods* 2012, **9(3):**245-253.
http://scholar.google.com/scholar?cluster=5929293844302699532&hl=en&as_sdt=0,50

[8] de Chaumont F, Dallongeville S, Chenouard N, Hervé N, Pop S, Provoost T, Meas-Yedid V, Pankajakshan P, Lecomte T, Montagner YL, Lagache T, Dufour A, Olivo-Marin J: **Icy: an open bioimage informatics platform for extended reproducible research.** *Nature Methods* 2012, **9(7):**690-696.
http://scholar.google.com/scholar?cluster=8170444028478407032&hl=en&as_sdt=0,50

[9] Carpenter AE, Jones TR, Lamprecht MR, Clarke C, Kang IH, Friman O, Guertin DA, Chang JH, Lindquist RA, Moffat J, Golland P, Sabatini DM: **CellProfiler: image analysis software for identifying and quantifying cell phenotypes.** *Genome Biology* 2006, **7:**R100.
http://scholar.google.com/scholar?cluster=11680963210515374635&hl=en&as_sdt=0,50

[10] **SciJava** [http://scijava.org/]

[11] **Maven** [http://maven.apache.org/]

[12] Preston-Werner T: The GitHub Effect: Forking Your Way to Better Code. [http://lanyrd.com/sfxcw]

[13] Carpenter AE, Kamentsky L, Eliceiri KW: **A call for bioimaging software usability.** *Nature Methods* 2012, **9(7):**666-670. http://scholar.google.com/scholar?cluster=3781196302537129054&hl=en&as_sdt=0,50

[14] **ImageJ source code** [https://github.com/imagej/imagej]

[15] **ImageJ public mailing list** [http://developer.imagej.net/mailing-lists]

[16] **ImageJ issue tracker** [http://trac.imagej.net/]

[17] **ImageJ roadmap** [http://trac.imagej.net/roadmap]

[18] **ImageJ pull requests** [https://github.com/imagej/imagej/pulls]

[19] Yoo TS, Ackerman MJ, Lorensen WE, Schroeder W, Chalana V, Aylward S, Metaxas D, Whitaker R: **Engineering and algorithm design for an image processing api: a technical report on itk-the insight toolkit.** *Studies in health technology and informatics* 2002, 586-592. http://scholar.google.com/scholar?cluster=10830740755452103531&hl=en&as_sdt=0,50

[20] Chiba S, Nishizawa M: **An easy-to-use toolkit for efficient Java bytecode translators.** In *Generative Programming And Component Engineering*. Berlin/Heidelberg: Springer; 2003:364-376. http://scholar.google.com/scholar?cluster=6256000376822757179&hl=en&as_sdt=0,50

[21] **SezPoz** [http://sezpoz.java.net/]

[22] **EventBus** [http://eventbus.org/]

[23] **KNIME Image Processing nodes** [http://tech.knime.org/community/image-processing]

[24] **Alida** [http://informatik.uni-halle.de/alida/]

[25] **Retrotranslator** [http://retrotranslator.sourceforge.net/]

[26] **Dropbox** [http://dropbox.com/]

[27] **List of ImageJ update sites** [http://fiji.sc/List_of_update_sites]

Illustrations and figures

[Figure 1: IJ1 case logic compared to a unified ImgLib2 implementation.]
https://www.dropbox.com/s/ca0ldacjjjoi6of/figure-1.png
Caption: On the left, we see the ImageJ1 implementation of a rolling ball background subtraction method, part of the ij.plugin.filter.BackgroundSubtracter class. On the right, we see an equivalent implementation using ImgLib2, without the need for extensive case logic.

[Figure 2: A mixed-world ImageJ1 + ImageJ2 command.]
https://www.dropbox.com/s/txgakdh6o719yfn/figure-2.png
Caption: This example ImageJ1 plugin utilizes ImageJ2 as a library to generate a spreadsheet-style results table with strings for cells, something normally not possible in ImageJ1.

[Figure 3: Comparison of pure ImageJ1 command with one using ImageJ2's declarative syntax.]
https://www.dropbox.com/s/fkcaj7eb4z2is0n/figure-3.png
Caption: On the left, we see an ImageJ1 implementation of a plugin that copies slice labels from one image to another, as chosen by the user. On the right, we see the same plugin written using ImageJ2's declarative command syntax. The actual operation (the copyLabels method) is identical, but the routine for selecting which images to process is no longer necessary.

[Figure 4: ImageJ2 in action.]
https://www.dropbox.com/s/lz259mtxpsnoc4z/figure-4.png
Caption: ImageJ2's default user interface is closely modeled after ImageJ1, so will be familiar to existing users. But the underlying architecture supports many alternative UIs as well.

Tables and captions

Table 1: Numeric types supported by ImageJ

Bit depth	**Signedness**	**Values**	**Min. Value**	**Max. Value**	**ImageJ1**	**ImageJ2**
1-bit	unsigned	binary	0	1	no	yes

8-bit	unsigned	integer	0	255	yes	yes
12-bit	unsigned	integer	0	4,096	no	yes
16-bit	unsigned	integer	0	65,536	yes	yes
32-bit	unsigned	integer	0	2^{32} - 1	no	yes
8-bit	signed	integer	-128	127	no	yes
16-bit	signed	integer	-32,768	32,767	no	yes
32-bit	signed	integer	-2^{31}	2^{31} - 1	no	yes
64-bit	signed	integer	-2^{63}	2^{63} - 1	no	yes
32-bit	signed	floating point	approx. -3.4 × 10^{38}	approx. 3.4 × 10^{38}	yes	yes
64-bit	signed	floating point	approx. -1.8 × 10^{308}	approx. 1.8 × 10^{308}	no	yes
3 × 8-bit	unsigned	integer	(0, 0, 0)	(255, 255, 255)	yes	legacy†
8-bit	indexed	integer	(0, 0, 0)	(255, 255, 255)	yes	legacy†

† ImageJ2 supports these pixel types only for backwards compatibility with ImageJ1, since composite color mode with one color lookup table (LUT) per channel achieves the same results in a more flexible way.

Table 2: Core plugin types

AutoThresholdMethod	An algorithm for use with automatic thresholding. The AutoThresholdMethod is an example of a new plugin type created for use with a specific command. Built-in examples include Huang, Otsu, and Percentile, but this plugin type makes it easy to extend ImageJ with additional thresholding algorithms.

CalculatorOp	A binary operation for the Image Calculator command. Like AutoThresholdMethod, the CalculatorOp is an example of a new plugin type created for use with a specific command. The SciJava Common plugin framework makes it easy for anyone to define these sorts of additional plugin types and easily discover them at runtime without excessive boilerplate code.
Command	The bread and butter plugin of ImageJ, defining an operation with typed inputs and outputs. Commands typically appear in the menu system of ImageJ's user interface, but can be exposed via interoperability mechanisms in many other ways, such as nodes in KNIME or modules in CellProfiler. When ImageJ users talk about "writing a plugin" they usually mean a command.
Display	A plugin for visualizing data. For example, an ImageDisplay shows two-dimensional planes of N-dimensional image data in a window with sliders for controlling which plane is visible. However, there are no limits on the sorts of objects that can be visualized: other examples include the TextDisplay, which shows strings, and the TableDisplay, which shows tabular data as a spreadsheet.
Platform	A collection of extensions for a specific platform (such as operating system type, version or architecture, or a particular version or implementation of Java). These plugins allow customization of ImageJ's behavior based on the platform; e.g., on OS X, the menu bar appears at the top of the screen, with the About, Preferences and Quit commands relocated to the Application menu.
PostprocessorPlugin	A "meta-command" which does something with a command after it has run. For example, the DisplayPostprocessor takes care of displaying the outputs of a command after it has completed execution. Postprocessor plugins are executed in priority order as part of the command "postprocessing chain" after the command is actually executed.
PreprocessorPlugin	A "meta-command" which prepares commands to run. For example, the LoadInputsPreprocessor populates a command's inputs with the last-used values as defaults, which can save the user a lot of time. Preprocessor plugins are executed in priority order as part of the command "preprocessing chain" before the command is actually executed.
ScriptLanguage	A programming language for the script editor. Each ScriptLanguage provides the logic necessary for ImageJ to support scripts written in that language (e.g., Javascript or Python). It also makes it possible for ImageJ to "record" commands executed as code snippets that can be reused in scripts to repeat those operations.

Service	A collection of related functionality, expressed as an Application Programming Interface (API). ImageJ services are singletons with respect to each application context. For example, each instance of ImageJ has exactly one AnimationService responsible for managing animations, providing methods to start and stop animations, select the dimension over which to animate, adjust frame rate, and other options.
Tool	A collection of rules binding user input (e.g., keyboard and mouse events) to display and data manipulation actions. For example, the PanTool pans a display when the mouse is dragged or arrow key is pressed; the PencilTool draws hard lines on the data within an image display.
Uploader	A protocol by which the ImageJ Updater can submit updated files to an update site on the Internet. Examples include SSH, SCP, SFTP and WebDAV.
UserInterface	An application user interface (UI). The flagship ImageJ user interface is written in Swing and modeled closely after the ImageJ 1.x design. But the design makes other UIs equally possible. Since a UI is simply a type of plugin, anyone can develop their own UI on top of ImageJ2 without any code changes to the core system. ImageJ2 is even flexible enough to display multiple simultaneous UIs at the same time.
View	A wrapper plugin for use with a Display. ImageJ maintains a strict separation of concerns between the data itself—such as an N-dimensional image—and the visualization thereof—such as which plane is currently being visualized. A View maintains such visualization-specific state, with Displays operating on Views rather than on the data directly. In this way, two Displays can be easily synchronized by operating on the same View of the same data, or kept separate by operating on two separate Views of the same underlying data, all without needing to keep expensive copies of the data itself.
Widget	A user interface element for harvesting typed inputs. Typically, these widgets are presented as part of a form in a dialog box which prompts the user to fill in input values of a command. But in principle the widgets can be used for anything requiring typed input from the user. For example, a FileWidget allows the user to select a file (java.io.File) on disk, while a ToggleWidget provides a boolean toggle (typically rendered as a checkbox). ImageJ provides UI-agnostic interfaces to the common widget types, along with widget implementations corresponding to each supported UserInterface plugin. However, an extension to ImageJ can not only implement its own

Chapter 6: Open Source Lifetime Analysis

Adapted from: SLIM Curve: An Open Source Analysis Library for Fluorescence Lifetime Microscopy. PR Barber*, A Grislis[†], CT Rueden[†], MC Hiner[†], MI Rowley[‡], J Nedbal[§], J Schindelin[†], B Vojnovic*, KW Eliceiri· PLOS ONE.

Summary

With the advent of improved labelling schemes, more sensitive detection and improved algorithms, the field of fluorescence microscopy has rapidly become very focused on precise quantitation of signals in space and time. With these improved technologies, and the biological problems that both utilize and drive them, there has arisen demand for technologies that can fully exploit all the information in the fluorescent pixel to best understand the dynamic interactions that occur at both the cellular and subcellular level. One imaging technique with proven ability for yielding additional information from the fluorescence experiment is Fluorescence Lifetime Microscopy (FLIM). FLIM can measure changes in the microenvironment such as pH and changes in protein conformational state to uniquely provide an image with correlated chemical information. FLIM has been rapidly adopted by the imaging community for exploitation in a myriad of biological studies and yet this increased use has exposed the dearth of computational framework approaches for the analysis and sharing of FLIM data. To address this need, we have developed SLIM Curve, a library based platform that serves the developer and end user alike in the development and application of robust solutions for FLIM analysis.

I. Introduction

In the last thirty years there has been an explosion in advanced biological imaging techniques that have allowed for an unprecedented ability to interrogate biological phenomena at cell and subcellular resolution. One of these powerful techniques has been modern fluorescence microscopy with key inventions such as the laser scanning microscope and green fluorescent protein. These imaging tools along with other enabling technologies have become essential tools for modern biological imaging. With these have come great interest in exploiting the full information in the pixel beyond fluorescence intensity, and taking advantage of spectra, polarization and lifetime. Fluorescence lifetime has been of particular interest due to its unparalleled ability to probe the microenvironment and be sensitive to such changes as Förster Resonance Energy Transfer (FRET), protein conformation, metabolism, pH and hydrophobic regions. Fluorescent lifetime has been in use for decades as cuvette-based measurement techniques but in the last twenty years, with the rise of improved electronics and powerful computation, Fluorescence Lifetime Imaging (FLIM) has developed in which time-correlated photon counts for every pixel can be registered as an image. This means that a direct correlation between physical structure and chemical microenvironment can now be made. When coupled with fluorescence tagging strategies, further information about function can be surmised, and when used with live cell imaging approaches, temporal information can be correlated. FLIM has grown rapidly from a technique only in the domain of instrumentalists to advanced turn-key systems used by bench biologists. FLIM is now widely used in a variety of applications from measuring the metabolic state of differentiating stem cells and intrinsic signatures of cancer cells to measuring changes in the lipid rafts of cancer and FRET of signaling events in cell division.

FLIM is available in two primary modes, time domain and frequency domain, but also is compatible with a number of different microscopy configurations ranging from widefield, confocal, spinning disc, to multiphoton microscopy. More recently several research groups are experimenting with super resolution FLIM and even endoscopy. This expanded use of FLIM has resulted in further innovation of the underlying FLIM technology including faster electronics and more sensitive detection. This has meant that shorter lifetimes can be measured, with more time channels collecting photons from weaker sources and over a range of wavelengths. Despite these great advances in biological application and instrumentation there has been surprisingly a lack in corresponding development in image informatics tools to directly support the FLIM imaging modality.

The great advances in FLIM have largely been enabled by computation and software. Improved software tools have allowed not only for the FLIM electronics to be robustly controlled and capture short lifetimes, but to do so rapidly so that FLIM images in 3D (space) and 4D (space and time, or space and spectral) can be collected (Studier *et al.*, 2014). Curve fitting algorithms have been developed that allow for robust fitting for two or more fluorescent components. These algorithms are based on the work of other fields including geosensing and astronomy and are implemented in a number of open and proprietary toolkits such as SciPy (http://www.scipy.org/), Boost (http://www.boost.org/), GSL (http://www.gnu.org/software/gsl/), MATLAB (Mathworks, Natick, MA, USA) and Globals (http://www.lfd.uci.edu/globals/). However these systems are not designed for the bench biologist to use and are usually most appropriate as tools for developers and engineers to use for their experimentation. Several companies have made commercial packages for FLIM analysis but these are closed source tools that are not transparent in their analysis and only support their own file formats, which makes sharing of approaches and FLIM data difficult. In recognition of the need for more transparent methods for FLIM analysis, there have begun to be developed new turn-key open analysis FLIM tools such as FLIMfit from the French group (Warren *et al*., 2013). However there needs to be a more complete informatics approach to meet specific and currently unmet needs. These include: 1) the need for a modular and flexible FLIM algorithm library that can be utilized and modified easily by a developer; 2) a turnkey FLIM analysis tool that uses the library and yet still can be used easily and directly by the bench biologist; 3) the integration of FLIM analysis with microscopy image analysis; 4) the support for all open and commercial FLIM file formats; 5) be done in a way that can be adapted for further integrated analysis of closely related fluorescence dimensions such as spectral and polarization.

We built on previous efforts in lifetime analysis (Barber *et al*., 2005, 2009) and spectral lifetime analysis (Rueden *et al*., 2004) to build a software library, SLIM Curve, and applications that use it, TRI2 (Barber *et al*., 2009) and a plugin for ImageJ (Schneider *et al*., 2012), in a FLIM analysis system that directly addresses these five needs. We recognized that current methods, such as those in MATLAB, may be difficult for the biological community to use. Further, many of these methods are also not attractive for developers because much of the published code, such as Numerical Recipes (Press *et al*., 1992), may have restrictive licenses and a steep learning curve. We sought to develop a software system that would complement current commercial efforts and in fact directly support the image acquisition and image processing formats of these systems. We decided to focus our initial efforts on time-resolved FLIM data such as that collected from the PicoQuant and Becker & Hickl hardware systems, but the library was designed to be flexible enough to support frequency domain in the future. Similarly, as the name of the library suggests, the goal is also to support dimensions that can be closely correlated, such as spectra and polarization. Currently the spectral support is minimal but this will be augmented as the library evolves. This flexibility is supported and can be exploited by the use of the ImgLib2 (http://imglib2.net/) data container, which, when used with the forthcoming ImageJ2 (http://developer.imagej.net/), allows for the support of data of unlimited dimensions, in principle, and will handle FLIM data that includes additional channels of spectra and polarization. As described below the SLIM Curve library can either be used standalone (i.e. from the command prompt); from third party applications such as MATLAB; from its own largely FLIM-specific application, TRI2; or from ImageJ as a plugin that can integrate FLIM analysis with image processing. A myriad of imaging file formats can be supported through the use of the SCIFIO (http://scif.io/) infrastructure to support a range of open and proprietary file formats. To our knowledge this is the first FLIM analysis system that offers this range of flexibility and functionality.

The approach to use a core open-source software library brings the usual advantages of open source that enable community contribution to, and verification of, the underlying code. It also enables the release of ‘polished’ graphical-user-interface-driven applications alongside a command line tool, and allows use in other environments. In the following section a description of the library is given, followed by how it can be extended for some more advanced uses. The additional code that builds the library into a software system with different methods of use is then demonstrated.

2. SLIM Curve: FLIM Analysis Library Description

2.1 Library Description

The SLIM Curve library offers standard non-linear least squares fitting in the form of the Levenberg-Marquardt (LM) algorithm and more advanced algorithms such as maximum likelihood and global analysis that has been optimized for FLIM (Barber *et al.*, 2009), as well as simpler methods such as the rapid lifetime determination (RLD) by integration (Sharman *et al.*, 1999), and decay curve exploration via the method of phasors (Digman *et al.*, 2008, Wouters and Esposito 2008). In particular, the library has the ability to account for an instrument response function (IRF, prompt or excitation function) that distorts the pure exponential decay. Through iterative re-convolution, the LM algorithm extracts the true lifetime from IRF distorted signals. The addition of new methods is also allowed and can inherit the standard library code interface, such as the Bayesian algorithms currently under development within this framework (Rowley *et al.*, 2011).

Written in pure ANSI C, the library remains cross platform compatible and can be compiled to run fast as a native executable, yet remains accessible to many high-level programmers using Java, Python, MATLAB, C++ and C#, to name a few. The intention is also that the user interaction with the library could be in a number of forms according to the user's choice. That is, interaction could be through a graphical user interface such as with TRI2 or ImageJ, or could be through the command line in a scriptable form, or could be via a third party framework such as MATLAB or R (http://www.r-project.org/). We demonstrate use of the library through the standalone Windows program TRI2, the Java-based cross-platform program ImageJ, and through the popular numerical mathematics program MATLAB, as well as a simple headless command line executable. Use of the library via a cloud or web service would also be possible with additional development.

Full details on SLIM Curve, including downloads, can be found on the project web site at:

https://slim-curve.github.io/

2.2 Library Contents

2.2.1 Data analysis methods

The following fitting and analysis methods are currently present in the open source library for lifetime data:

- **Rapid Lifetime Determination (RLD)**

  A method based on 3 integrals to determine a single average lifetime (Sharman *et al.*, 1999), including a variant that accounts for the IRF.

- **Levenberg-Marquardt (LM) Non-Linear Least Squares**

  A classical LM algorithm (Levenberg 1944), the performance of which is modified by the noise model (see 2.2.2). Multi-exponential and stretched exponential analyses are built in, others can be added, as are parameter fixing and restraining.

  There are variants with and without an IRF. Parameter error estimates are returned based on the fitting alpha matrix (Press *et al.*, 1992)

- **Maximum Likelihood Estimation**

  This variant of the LM algorithm is accessed by using a specific noise model (see 2.2.2).

- **Phasor**

  Transformation to phasor space for the calculation of a single average lifetime and graphical multi-exponential analysis (Wouters and Esposito 2008)

Primarily for spectral data, a non-negative linear unmixing algorithm is included (Barber et al., 2003), which can be used to resolve known spectral components from a mixture.

In addition, the development of Bayesian algorithms (Rowley *et al.*, 2011) is currently closed source but this functionality is included in the TRI2 executable. These additional algorithms allow multi-exponential resolution and model selection from photon evidence based analysis. IRF determination from the exponential decay data is also possible.

2.2.2 Noise Models

Several noise models can be chosen to influence the LM optimization via the chi squared parameter.

- 'Constant' - every data point is assumed to have the same supplied variance
- 'Given' - every data point can have an individual variance, given via a data array
- 'Gaussian' - Variance for Gaussian distribution is used at all data points. Variants based on the data or the fit.
- 'Poisson' - Variance for Gaussian distribution is used with a lower limit of 15, this being the point where the Gaussian approximation begins to break down with Poissonian data. Variants based on the data or the fit.
- 'MLE' - Maximum likelihood estimation through the use of the Poisson equation (Laurence and Chromy, 2010)

2.3 Extensions

Based on the features of the library explained above, several important extensions are readily possible and these are detailed below.

2.3.1 Global Analysis

In some biological experiments it is extremely advantageous to analyze data from a whole image or series of images simultaneously to determine certain parameters with a high accuracy yet allow other parameters to remain variable at the pixel level to capture spatial variations. In this so called 'global analysis', some parameters can be globally determined whilst others remain local (Verveer *et al.*, 2000, Pelet *et al.*, 2004). This is particularly useful when a biosensor based on FRET is in use (Parsons et al., 2005) which usually exists in either an activated or a deactivated state, that are represented by different measured lifetimes by FLIM. Thus, these characteristic lifetimes can be determined globally, whilst the determination of the fraction of activated molecules can be determined locally (Barber *et al.*, 2009). The library has built in optimized functions for this type of analysis (Barber *et al.*, 2005), that offer fast convergence and built-in optimization of initial parameter estimates. Methods for global analysis involving other generic functions (e.g. an exponential rise or non-exponential functions) are also included in the library.

2.3.2 Support Plane Analysis, 1D and 2D

When initially exploring new data it is sometimes desirable to investigate whether the optimized solution, determined by LM, is well confined or whether other, equally likely, nearby parameter solutions exist. This can be done be exploring the chi squared values around the minimum. The library offers built-in functions to explore this ‘support plane’ by determining the chi squared value for a range of parameter values determined by the user. This can be performed for 1 parameter (1D) or 2 parameters (2D) simultaneously and is exploited by TRI2 to plot a 3D graph the chi squared around the minimum.

2.3.3 Chi squared based fitting model selection

When fitting data it is often not clear what fit model should be used to try and explain the data and several techniques exist to help the user determine if the selected model is a good fit to the data. The user can examine the fit overlaid on to the data, or examine the residuals (the difference between the data and the fit), or examine the autocorrelation of the residuals (Lakowicz, 1999). The two latter methods should result in a random distribution of data, and any structure in the data that deviates from this represents systematic errors characteristic of a bad fitting model. The user is recommended to use these techniques.

However, we have implemented in TRI2 an automated algorithm based on the optimized chi squared value that can help choose between two fitting models. Bayesian model selection may prove to be the gold standard method in this respect (paper in preparation), but such methods are computationally intensive and a good, fast approximation can be made with a simple calculation based on chi squared values.

We propose that the chi squared difference test for nested models (Schermelleh-Engel et al., 2003) can be used to distinguish between multi-exponential models. This is valid when one model is a simplification of another (nested within), and in the case of FLIM we can choose a mono-exponential model as a simplification of a bi-exponential model as follows,

$$\mathrm{I_{mono}(t) = Ae^{-\frac{t}{\tau}} + Z,}$$

$$\mathrm{I_{bi}(t) = A_1e^{-\frac{t}{\tau_1}} + A_2e^{-\frac{t}{\tau_2}} + Z.}$$

The difference in degrees of freedom between the two models is 2, since the more complicated model has 2 more free parameters. Using a table of chi squared values, the p<0.05 significance test for 2 degrees of freedom will pass when the difference in chi squared is greater than 5.99 (Toutenburg 1971). The SLIM Curve library returns the *raw* chi squared value from each LM fit. It is often more convenient to consider the *reduced* chi squared value which can be calculated from the raw value by dividing by the total degrees of freedom (this is equal to the number of time bins minus the number of free parameters).

Consider an example with 256 time bins and in which a bi-exponential model is a good fit and returns a reduced chi squared value of 1.00, indicating a very good fit. The chi squared value returned by a mono-exponential fit must be significantly higher for us to reject it and accept the bi model as the most likely explanation of the data. To reject the mono model we require a difference in raw chi squared value of

$$\chi^2_{\mathrm{diff}} = \chi^2_{\mathrm{r,mono}}(256 - 3) - \chi^2_{\mathrm{r,bi}}(256 - 5) \geq 5.99.$$

Which translates into a limit on the reduced mono chi squared value of

$$\chi^2_{\mathrm{r,mono}} \geq 1.016,$$

indicating that the mono-exponential fit must have reduced chi squared value of greater than 1.016 for us to accept the bi-exponential model. An example of the application of model selection can be found in Figure 1.

3. Using the SLIM Curve library

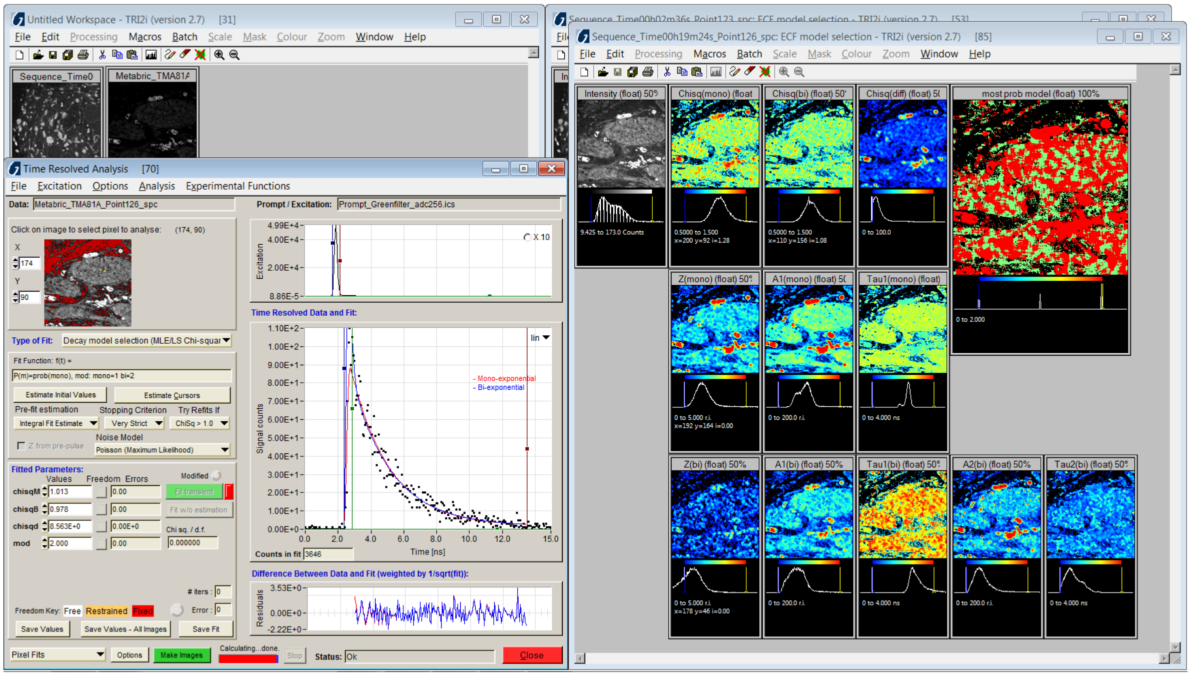
3.1 Use with a graphical user interface in the TRI2 program

The Time Resolved Imaging version 2 (TRI2) computer program was first released to selected researchers in 2004, since then it has been continually developed, increasing in functionality, and from it the SLIM Curve library was extracted. TRI2 is a closed source application that uses identical SLIM Curve code to that released under the open source license (as copyright holders, we have those rights). It is written in C, uses the user interface framework from LabWindows/CVI (version 8.0.1, National Instruments Inc.) and handles images internally with the FreeImage library (http://freeimage.sourceforge.net/). As such, the SLIM Curve library is directly compiled into the TRI2 executable.

TRI2 provides image analysis capabilities to the time-resolved transient fitting of SLIM Curve. Once a time resolved image has been loaded into TRI2, either from an .ICS file (http://libics.sourceforge.net/), e.g. from a Gray Open Microscope (Barber et al., 2013), or from one of several types of Becker & Hickl .SDT files, access to the SLIM Curve library is provided by a dedicated time resolved (TR) user interface panel. Figure 1 shows a screenshot of several windows of TRI2. The “Time Resolved Analysis” window shows data from one pixel bin and to the right is a window with fitting results from a whole image.

Access to the different fit types of the library and to the setup parameters are easily accessible through user interface components, such as loading an IRF for repeated use. The real advantage of such a program is the ability to perform pre- and post-processing of images and the presentation of starting images and results to the user for examination.

TRI2 also contains many convenience features and ‘macros’ that make image processing for specific tasks easier. General functions such as arithmetic, file conversion and re-sampling can be coupled with more specific functions such as reconstructing an image from a laser scanning microscope with a bi-directional scanner. Macros for the measurements usual in FLIM are included, e.g. for fractional contribution, fractional intensity and average lifetime. Novel developments, such as the Bayesian analysis routines, are often worked into TRI2 in a usable form before anywhere else, such that they can be tested in batch and automated procedures. This is usually in an internal development version of TRI2. One such development was the multi-exponential lifetime filter that is useful when extracting the lifetime of a fluorophore from unwanted and heterogeneous environment, such as a human biopsy tissue section (Barber *et al.,* 2014). There, we showed that areas consisting of a largely mono-exponential response in a given range, can be extracted from a multi-exponential environment in a way that is independent of intensity and the average lifetime. The other extensions described in this paper, global analysis, support plane analysis and model selection are also included in the TRI2 build.


Untitled Workspace - TRI2i (version 2.7) [31]

Time Resolved Analysis [70]

Data: Metabric_TMA81A_Point126_spc

Prompt / Excitation: Prompt_Greenfilter_adc256.ics

Click on image to select pixel to analyse: (174, 90)

X 174

Y 90

Type of Fit: Decay model selection (MLE/LS Chi-squar

Fit Function: f(t) = P(m)=prob(mono), mod: mono=1 bi=2

Pre-fit estimation: Integral Fit Estimate

Stopping Criterion: Very Strict

Try Refits If: ChiSq > 1.0

Noise Model: Poisson (Maximum Likelihood)

Fitted Parameters:

	Values	Errors
chisqM	1.013	0.00
chisqB	0.978	0.00
chisqd	8.563E+0	0.00E+0
mod	2.000	0.00

Chi sq. / d.f. 0.000000

Counts in fit 3646

iters : 0

Error : 0

Calculating...done.

Status: Ok

Sequence_Time00h02m36s_Point123_spc: ECF model selection - TRI2i (version 2.7) [53]

Sequence_Time00h19m24s_Point126_spc: ECF model selection - TRI2i (version 2.7) [85]

Figure 1, A screenshot of the TRI2 program performing exponential model selection. The user interface window to the lower left is used to control and examine individual fits to the data from pixel bins. Two fits (read and blue lines) are shown in the main large plot, with the transient data (black dots), the IRF plot is above and the residuals plot below. On the right hand side is the result of applying the analysis to a whole image. Several parametric images are created with the image to the top-right representing the model selected at each pixel; red and green represent bi- and mono-exponential, respectively. More than one image may be processed, creating multiple results windows. The original FLIM image is of a section of human breast cancer biopsy tissue, stained for the protein Her3 with the Alexa 546 fluorophore, acquired on a Gray Open Microscope (Barber *et al.*, 2013).

Figure 1 represents the processing of 44550 pixels (image size is 256 by 256, low intensity pixels have been excluded) in 56.7 seconds on a quad-core intel i7 processor, with model selection requiring 2 fits per pixel. This equates to approximately 5-10 ms per maximum likelihood fit when SLIM Curve is natively compiled on a Windows 32-bit platform.

In addition to the FLIM *image* based TRI2 program, a "Single Transient Import" companion program wraps around SLIM Curve to provide fitting of single transient curves in a similar user interface. Data import is eased through the ability to directly copy a dataset from popular computer spreadsheet programs. Both TRI2 and Single Transient Import are available from the University of Oxford website (http://users.ox.ac.uk/~atdgroup/software).

Being custom and closed programs, these applications can be optimized for a single operating platform and for particular tasks, but they lack the power of third party contributed code and additional flexibility of an open source program such as ImageJ. Integration of the SLIM Curve library into the ImageJ platform is the subject of the next section. TRI2 and similar programs do benefit, however, from the open source nature of the SLIM Curve library in code robustness that comes from diverse use (by different programmers and in different use cases) and in openness in that the underlying algorithms can be examined and verified.

3.2 Use the SLIM Curve plugin for ImageJ

Since 2010 we have been developing a plugin for ImageJ which provides a graphical user interface around the SLIM Curve fitting engine.

3.2.1 Installing the plugin

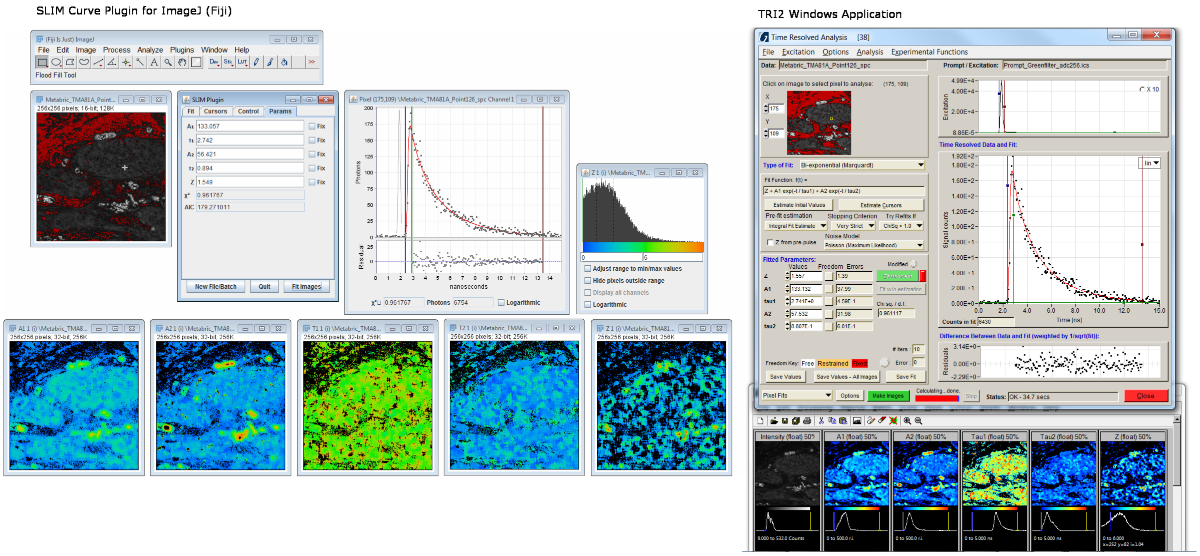
The SLIM Curve plugin for ImageJ (http://fiji.sc/SLIM_Curve) is available via an ImageJ update site. The simplest way to install it is to download the Fiji distribution of ImageJ for the life sciences (Schindelin *et al.*, 2012, http://fiji.sc/), and enable the SLIM Curve update site as described on the Fiji wiki (http://fiji.sc/Update_Sites). Once installed, the plugin is accessible from the Lifetime submenu of ImageJ's Analyze menu. It can also be launched quickly using ImageJ's Command Finder by pressing the L key and typing "slim" into the search box.

3.2.2 Supported file formats

Once launched, the SLIM Curve plugin will prompt for a lifetime source image to open. The plugin uses the SCIFIO and Bio-Formats (Linkert *et al.*, 2010, http://openmicroscopy.org/info/bio-formats) libraries for scientific image I/O, which together provide support for more than 130 life science file formats. In particular, we have thoroughly exercised the plugin with ICS files produced at the Gray Institute, as well as Becker & Hickl SDT files produced by both Gray and LOCI acquisition systems.

3.2.3 Current functionality

The SLIM Curve plugin exposes many of SLIM Curve's configuration options in its main control window. It can fit individual pixels, entire images per-pixel, or perform single fits for entire images, using SLIM Curve's RLD, LM or hybrid (RLD+LM) fitting algorithms. It supports single, double and triple exponential fits, as well as stretched exponential fits. Available noise models include Gaussian, Poisson and MLE as discussed in section 2.2.2 above. The plugin produces one or several fitted images depending which parameters (A, $\boldsymbol{\tau}$**,** Z, χ^2) are chosen for visualization. It also offers full control over the start and end fit cutoffs known as “cursors”; several binning options to reduce noise when fitting per-pixel; support for so-called “excitation” or “prompt” files containing a recorded system response function to be convolved with the exponential fit; the ability to export numerical results to text format for further analysis outside ImageJ; and batch processing support for analyzing many lifetime images as part of an ImageJ macro or scripting workflow. Lastly, note that we have extensively compared the statistics produced by TRI2 and the SLIM Curve plugin to ensure that either one can be used with the same end result (Figure 2). Full usage instructions for the plugin can be found on its web site on the Fiji wiki at http://fiji.sc/SLIM_Curve.



A₁ 133.057 □ Fix
τ₁ 2.742 □ Fix
A₂ 56.421 □ Fix
τ₂ 0.894 □ Fix
Z 1.549 □ Fix
χ^2 0.961767
AIC 179.271011

Type of Fit: Bi-exponential (Marquardt)
Fit Function: f(t) = Z + A1 exp(-t / tau1) + A2 exp(-t / tau2)
Pre-fit estimation: Integral Fit Estimate
Stopping Criterion: Very Strict
Try Refits If: ChiSq > 1.0
□ Z from pre-pulse
Noise Model: Poisson (Maximum Likelihood)

Fitted Parameters:

	Values	Errors
Z	1.557	1.39
A1	133.132	37.99
tau1	2.741E+0	4.59E-1
A2	57.532	31.98
tau2	8.807E-1	6.01E-1

Chi sq. / d.f. 0.961117
# iters : 10
Error : 0
Freedom Key: Free Restrained Fixed
Pixel Fits
Calculating...done.

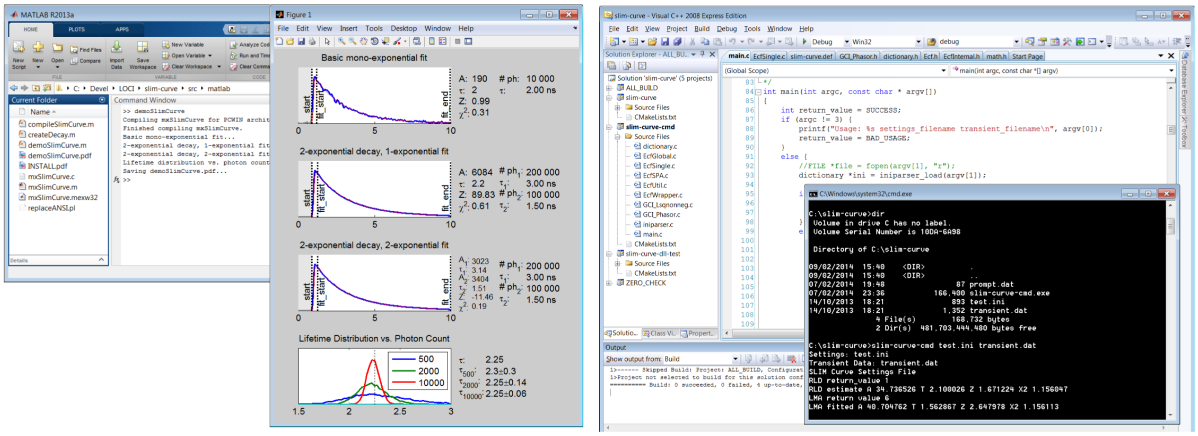
Figure 2, The ImageJ SLIM Curve plugin (Left side), showing the fitting of a single transient and the fitted parameters, with the parametric images resulting from fitting the whole image shown below. TRI2 fitting the same image (right side) for comparison.

3.3 Use from within MATLAB

The library includes a script file for MATLAB that compiles the library into a ‘mex’ file for use with Linux, MacOS and Windows operating systems, directly from the C code. Once this is complete the library can be called via a single call to “MXSLIMCURVE”. This allows the integration of the library with MATLAB scripts allowing this type of fitting with open source algorithms. Although MATLAB includes methods for curve fitting in the Curve Fitting and Optimization Toolboxes these generally use the method of least squares and as such are not optimized for data with Poissonian statistics such as photon count data. A demonstration script is included that will compile and then run the fitting routines, plotting the results as MATLAB graphs (see Figure 3).

3.4 Use as a standalone executable

As the library is written in C, compilation to a standalone executable is simple with the provided example “main.c” file and *CMakeLists* file. With this implementation the data and settings must be provided via text files, the names of which are passed to the executable. The settings file is designed to be compatible with TRI2 such that fitting types can be explored in TRI2 and then the settings saved for use at the command line. Figure 3 demonstrates compilation and execution under the Windows operating system.



```
>> demoSlimCurve
Compiling mxSlimCurve for PCWIN archit
Finished compiling mxSlimCurve.
Basic mono-exponential fit...
2-exponential decay, 1-exponential fit
2-exponential decay, 2-exponential fit
Lifetime distribution vs. photon count
Saving demoSlimCurve.pdf...
>>
```

```
 */
int main(int argc, const char * argv[])
{
    int return_value = SUCCESS;
    if (argc != 3) {
        printf("Usage: %s settings_filename transient_filename\n", argv[0]);
        return_value = BAD_USAGE;
    }
    else {
        //FILE *file = fopen(argv[1], "r");
        dictionary *ini = iniparser_load(argv[1]);

        i


    }
    e
```

```
1>------ Skipped Build: Project: ALL_BUILD, Configurat
1>Project not selected to build for this solution conf
========== Build: 0 succeeded, 0 failed, 4 up-to-date,
```

```
C:\slim-curve>dir
 Volume in drive C has no label.
 Volume Serial Number is 10DA-6A98

 Directory of C:\slim-curve

09/02/2014  15:40    <DIR>          .
09/02/2014  15:40    <DIR>          ..
07/02/2014  19:48                87 prompt.dat
07/02/2014  23:36           166,400 slim-curve-cmd.exe
14/10/2013  18:21               893 test.ini
14/10/2013  18:21             1,352 transient.dat
               4 File(s)        168,732 bytes
               2 Dir(s)  481,703,444,480 bytes free

C:\slim-curve>slim-curve-cmd test.ini transient.dat
Settings: test.ini
Transient Data: transient.dat
SLIM Curve Settings File
RLD return_value 1
RLD estimate A 34.736526 T 2.100026 Z 1.671224 X2 1.156047
LMA return value 6
LMA fitted A 40.704762 T 1.562867 Z 2.647978 X2 1.156113
```

Figure 3, Use of the SLIM Curve library in MATLAB (left) and as a headless command line program (right) compiled in Visual Studio 2008 C++ Express Edition (Microsoft Inc., WA, USA).

4. Discussion

In this paper we presented SLIM Curve, the open source C-code library for curve fitting and analysis of lifetime and spectral responses. We demonstrate how it can be used in different code environments including compilation into a native application, TRI2; being wrapped into an ImageJ plugin; compilation into a mex file for use with MATLAB scripts; as well as compilation into a headless and cross-platform command line application. Example code is included in the library. The use of the library is equally possible from other languages such as Python or from the R statistics package, for example.

The library includes a range of fitting routines for lifetime data based on Levenberg-Marquardt and maximum likelihood methods as well as analysis tools such as the phasor method and rapid lifetime determination from the area under the curve. The name 'SLIM' is derived from the phrase, “Spectral Lifetime Imaging Microscopy”, and highlights the intention that this library should not be restricted to lifetime but includes algorithms aimed at spectral analysis, such as the non-negative least squares algorithm for spectral unmixing (Barber *et al.*, 2003), already included. Future work will be aimed at the simultaneous analysis of spectral, lifetime and polarization information that is now routinely captured in some laboratories (Rueden *et al.*, 2004, Yan et al., 2006, Hunt et al., 2012). This is an area in which novel advanced analysis is greatly needed, especially as photon numbers from biological samples are limited (Zhao *et al.*, 2011) and the addition of more dimensions of measurement (i.e. time, spectral, polarization) can result in a very small number of photon counts per measurement channel.

The described analysis tools enable a wide range of FLIM experiments. Such as the simple detection of a change in lifetime due to a change in chemical environment (e.g. Gerritsen *et al.*, 1997), or viscosity changes (Kuimova *et al.*, 2008). As well as more advanced experiments such as the detection of FRET. Biologically useful extensions are also possible based on these core algorithms such as global analysis to increase signal to noise in lifetime invariant systems, support plane analysis to determine ‘confidence’ in fitted parameters, and model selection to help selection of the most appropriate model for the data.

Much of the future work will also focus on extending the functionality within ImageJ and in particular the Fiji distribution of ImageJ for the life sciences. For some time we have been developing a next generation version of ImageJ called ImageJ2 that greatly improves ImageJ’s support for n-dimensional scientific images; ImageJ2 now serves as the core architecture for all new development in Fiji: the current version of Fiji has all the functionality of classic ImageJ and supports all legacy work, yet within this program lies a new data engine that leverages the ImgLib2 project to support data of n-dimensionality. ImageJ2 also provides a new architecture for improved flexibility and modularity, including the ability to run Fiji “headlessly” from other applications or on a server cluster. Much of this development had in mind emerging applications such as FLIM, which combine spatial and non-spatial data analysis with the need to work with very large datasets. As FLIM continues to be adopted and used by the bench biologists and used in complex experiments, there will be demand to view it simply as one more channel in fluorescence microscopy experiments that can be analyzed as part of the entire data set, and not just as a piece on its own. This will require full access to FLIM analysis functions directly alongside imaging processing routines, which is something Fiji with the SLIM Curve library can uniquely do. We have developed a proof-of-concept ImageJ2-based version of the SLIM Curve plugin; future work will focus on further utilization of the ImageJ2 + Fiji platform, such as improved support for scripting SLIM analysis functions. As well, the SLIM Curve project can directly take advantage of ongoing developments in Fiji such as efforts to connect Fiji to the Open Microscopy Environment database system (http://www.openmicroscopy.org/), and the use of SCIFIO and Bio-Formats to open up proprietary microscopy file formats. These systems all boast facilities for metadata capture and display, and may also present opportunities for improved use of FLIM metadata in analysis and dissemination. Lastly, as Fiji continues to be optimized for speed and performance, and explores parallelization and possible GPU based applications, these are areas where improved FLIM analysis performance can be evaluated as well.

Acknowledgements

This work was supported by Cancer Research UK (grant C133/A/1812), EPSRC/BBSRC (grants EP/C546105/1 and EP/C546113/1), National Institutes of Health (grants GM092519 and CA136590) and National Science Foundation (1148362). We acknowledge input and advice from members of both the LOCI and Vojnovic labs for input and testing, especially Dr. Julian Gilbey, formerly of the Vojnovic Lab, for starting the library now known as SLIM Curve, in which many of the original features still remain. We are also indebted to Dr. Gregory Weitsman and the group of Prof. Tony Ng (King's College London, UK) for the preparation of biological samples of cells and tissues.

References

Barber, P R, Vojnovic, B., Atkin, G., Daley, F. M., Everett, S. A., Wilson, G. D. and Gilbey, J. D. (2003) Applications of Cost-effective Spectral Imaging Microscopy in Cancer Research. *Journal of Physics D: Applied Physics* **36**(14), 1729.

Barber, P. R., Ameer-Beg, S. M., Gilbey, J., Edens, R. J.,Ezike, I. and B. Vojnovic, B. (2005) Global and Pixel Kinetic Data Analysis for FRET Detection by Multi-photon Time-domain FLIM. *Proc. SPIE* **5700**, 171–181.

Barber, P. R., Ameer-Beg S. M., Gilbey J., Carlin, L. M., Keppler, M., Ng, T. C. and Vojnovic, B. (2009) Multiphoton Time-domain FLIM: Practical Application to Protein-protein Interactions Using Global Analysis. *J. R. Soc. Interface* **6**, S93–S105.

Barber, P.R., Tullis, I.D.C., Pierce, G.P., Newman, R.G., Prentice, J., Rowley, M.I., Matthews, D.R., Ameer-Beg, S.M. and Vojnovic, B. (2013) The Gray Institute 'open' high-content, fluorescence lifetime microscopes. *J. Microscopy* **251**(2), 154–67.

Barber, P. R., Tullis, I. D. C., Rowley, M. I., Martins, C. D., Weitsman, G., Lawler, K., Coffey, M., Woodman, N., Gillett, C. E., Ng, T. & Vojnovic, B. (2014). The Gray Institute open microscopes applied to radiobiology and protein interaction studies. *Proc. SPIE* 8949, 89490D–89490D–17.

Carpenter, A. E., Kamentsky, L. & Eliceiri, K. W. (2012). A call for bioimaging software usability. *Nature Methods* 9, 666–670.

Digman, M. A., Caiolfa, V. R., Zamai, M., and Gratton, E. (2008) The Phasor Approach to Fluorescence Lifetime Imaging Analysis. *Biophys. J.* **94**(2), L14–16. doi:10.1529/biophysj.107.120154.

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. *Nature Methods* 9, 697–710.

Gerritsen, H.C., Sanders, R., Draaijer, A., Ince, C. and Levine, Y.K. (1997) Fluorescence lifetime imaging of oxygen in living cells. *Journal of Fluorescence* **7**(1), 11–15.

Hunt, J., Keeble, A. H., Dale, R. E, Corbett, M. K., Beavil, R. L., Levitt, J., Swann, M. J., *et al.* (2012) A Fluorescent Biosensor Reveals Conformational Changes in Human

Immunoglobulin E Fc. *The Journal of Biological Chemistry* **287**(21), 17459–17470. doi:10.1074/jbc.M111.331967.

Kuimova, M.K., Yahioglu, G., Levitt, J.A. and Suhling, K. (2008) Molecular Rotor Measures Viscosity of Live Cells via Fluorescence Lifetime Imaging, *Journal of the American Chemical Society* **130**(21), 6672–6673.

Lakowicz, J. R. (1999) *Principles of Fluorescence Spectroscopy*. Vol. 2nd Edition. Plenum Publishers, New York.

Laurence, T. A. and Chromy, B. A. (2010) Efficient Maximum Likelihood Estimator Fitting of Histograms. *Nat Meth* **7**(5), 338–339.

Levenberg, K. (1944) A Method for the Solution of Certain Non-Linear Problems in Least Squares. Appl. Math. 2, 164–168.

Linkert, M., Rueden, C. T., Allan, C., Burel, J.-M., Moore, W., Patterson, A., Loranger, B., Moore, J., Neves, C., MacDonald, D., Tarkowska, A., Sticco, C., Hill, E., Rossner, M., Eliceiri, K. W. & Swedlow, J. R. (2010). Metadata matters: access to image data in the real world. *Journal of Cell Biology* 189, 777–782.

Parsons, M., Monypenny, J., Ameer-Beg, S. M., Millard, T. H., Machesky, L. M., Peter, M., Keppler, M. D. *et al.* (2005) Spatially Distinct Binding of Cdc42 to PAK1 and N-WASP in Breast Carcinoma Cells. *Molecular and Cellular Biology* **25**(5), 1680–1695.

Pelet, S., Previte, M. J. R., Laiho, L. H. & So, P. T. C. (2004). A fast global fitting algorithm for fluorescence lifetime imaging microscopy based on image segmentation. *Biophysical Journal* 87, 2807–2817.

Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T. (1992) Numerical Recipes in C: The Art of Scientific Computing, Cambridge University Press.

Rowley, M., Barber, P. R., Coolen, A. C. C. and Vojnovic, B. (2011) "Bayesian Analysis of fluorescence Lifetime Imaging Data." *Proc. SPIE* **7903**, 74.

Rueden, C., Eliceiri, K. W. and White, J. W. (2004) "VisBio: A Computational Tool for Visualization of Multidimensional Biological Image Data." *Traffic* **5**(6), 411–417.

Schneider, C. A., Rasband, W. S. & Eliceiri, K. W. (2012). NIH Image to ImageJ: 25 years of image analysis. *Nature Methods* 9, 671–675.

Schermelleh-Engel, K., Moosbrugger, H. and Müller, H. (2003) "Evaluating the Fit of Structural Equation Models: Tests of Significance and Descriptive Goodness-of-Fit Measures." *Methods of Psychological Research* **8**(2), 23–74.

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. *Nature Methods* 9, 676–682.

Sharman, K.K., Periasamy, A., Ashworth, H., Demas, J. N., and Snow, N. H. (1999) Error Analysis of the Rapid Lifetime Determination Method for Double-Exponential Decays and New Windowing Schemes. *Anal. Chem.* **71**(5), 947–952.

Studier, H., Weisshart, K., Holub, O. & Becker, W. (2014). Megapixel FLIM. *Proc. SPIE* 8948, 89481K–89481K–9.

Toutenburg, H. (1971) Fisher, R. A., and F. Yates: Statistical Tables for Biological, Agricultural and Medical Research. 6th Ed. Oliver & Boyd, Edinburgh and London 1963. X, 146 P. Preis 42 s Net. *Biometrische Zeitschrift* **13**(4), 285–285. doi:10.1002/bimj.19710130413.

Verveer, P. J., Squire, A. and Bastiaens, P. I. H. (2000) Global Analysis of Fluorescence Lifetime Imaging Microscopy Data. *Biophys. J.* **78**(4), 2127–2137.

Warren, S.C., Margineanu, A., Alibhai, D., Kelly, D.J., Talbot, C., Alexandrov, Y., Munro, I., Katan, M., Dunsby, C., et al. (2013) Rapid Global Fitting of Large Fluorescence Lifetime Imaging Microscopy Datasets. *PLoS ONE* **8**(8), e70687.

Wouters, F. S., and Esposito, A. (2008) "Quantitative Analysis of Fluorescence Lifetime Imaging Made Easy." *HFSP J.* **2**(1), 7–11.

Yan, L., Rueden, C. I., White, J. G., and Eliceiri, K. W. (2006) Applications of Combined Spectral Lifetime Microscopy for Biology. *Biotechniques* **41**(3), 249-257.

Young, P. A., Grislis, A., Barber, P. R., Keely, P. J. & Eliceiri, K. W. (2013). Data Processing for Time-Domain Fluorescence Lifetime Imaging Microscopy. *Microscopy and Microanalysis* 19, 758–759.

Chapter 7: HyperSpectral Analysis of confocal data.

Adapted from: Hyperspectral Multipoint High-Speed Confocal Microscopy. Andreas Velten, Jimmy J. Fong, John G. White, William Vogt, Jessica Plavicki, Matthew E. Larson, Jayne M. Squirrell, Elizabeth A. Harvie, Anna Huttenlocher, William M. Bement, Thomas R. Mackie, and Kevin W. Eliceiri. Journal of Microscopy.

Summary: Live cell imaging of biological phenomena often necessitates the combination of high-speed acquisition with high spatial resolution while tracking multiple fluorescently labeled components. This need has driven many advances in optical microscopy including the development of improved spectral detectors, optical sectioning techniques, high speed imaging, and sophisticated analysis approaches that can isolate fluorescence signals of interest from interfering background in space and time. We report the development of a hyperspectral confocal microscope that leverages the high-speed, high sensitivity capabilities of multipoint confocal designs with an Amici prism dispersing element and spectral analysis software. This hyperspectral, multipoint scanner can image multiple spectral channels simultaneously at high-speed while retaining high spatial resolution and sensitivity.

Introduction

Over the last thirty years laser scanning microscopy (LSM) has become an indispensable tool in modern biological research. This is largely due to LSM's unrivaled ability to provide high-resolution fluorescence optical sections of *in vivo* structures together with their dynamics. Fluorophores can range from endogenous metabolites (e.g. reduced nicotinamide adenine dinucleotide, NADH) and genetically engineered proteins (e.g. GFP::tubulin) to exogenous probes (e.g. FM series lipid probes or the Ca++ indicator Fura). LSM techniques such as confocal laser scanning microscopy (CLSM) (White and Amos 1987; Carlsson et al., 1985; Paddock, 1999; Sheppard et al., 1997) and multiphoton laser scanning microscopy (MPLSM)(Denk et al., 1990; Helmchen and Denk, 2005; Zipfel et al., 2003) microscopy have been developed to provide images from within a solid specimen that are largely free of interference from out-of-focus signals arising from fluorescent structures above and below the plane of focus. Stacks of optical sections can be collected enabling the three-dimensional structures of living specimens to be visualized over time (Carlsson et al., 1985; Saggau, 2006). Development of fast confocal scanning technologies such as multipoint, spinning disc (Lam et al., 2014; Petran et al., 1986) and slit scanners (Auran et al., 1995) have allowed high speed image collection of rapid biological events such as calcium transients in neuronal signaling (Greinberger and Konnerth, 2012). Both non-confocal, push-broom (Nagaoka et al., 2012; Schultz et al., 2001; Sinclair et al., 2004) and confocal (Owen et al., 2007) slit scan systems have been used with spectral detection. These are relatively slow as only one column (or row) of pixels is captured at a time (the other axis being used for spectral information). The non-confocal systems provide little out-of-focus rejection. The confocal slit systems only provide rejection of out-of-focus light that is proportional to the distance from the focal plane.. In contrast, point-scanning confocals achieve rejections proportional to the square of the distance. This renders slit scanners less suited for imaging at depth in scattering specimens. Slit scanning systems also may not be not be ideal when used with a square EMCCD or CMOS sensor typically used on a microscope. In a standard slit scanning setup this leads to a resolution in the spectral axis equal to the resolution in both spatial axes and consequently a poor spatial resolution or a an excessive spectral resolution resulting in slow data collection. Unfortunately, despite advances in speed and detection, fast confocals are currently limited when dealing with multiple fluorophores, particularly those with overlapping spectra. Many approaches have been developed for spectral detection with confocal microscopes (Bertani et al., 2013; Bini et al., 2011; Dickinson et al., 2001; Lerner et al.; Zucker et al., 2007), most using a filter or grating technology. However, these approaches are often light inefficient and can slow the imaging due to either longer collection times needed for acquisition of sufficient light, or to exchange optical filters. This is a major problem when imaging multiple fluorophores in biological specimens where rapid capture of events is required. This issue and the large number of physiological reporters now available and the frequent requirement to observe the dynamics of biological phenomena, all underscore the need for high-speed confocal imaging that provides spectral resolution with negligible sensitivity penalties.

Uses of color discrimination in optical microscopy

Spectral information is critical in microscopy to obtain functional information when examining endogenous or exogenous fluorescent reporters (Lavis and Raines, 2008; Pinaud et al., 2006; Rizzo et al., 2009; Zimmermann, 2005). Efficient discrimination of spectral signals is important for isolating a fluorescence signal of interest from background, be it intrinsic fluorescence such as NADH in metabolism studies, or an extrinsic label such as GFP-labeled signaling proteins in a subcellular trafficking study. Spectral analysis can be used not only to establish the identity of a fluorophore but also to study more complex phenomena, such as the spectral shifts produced by fluorescence resonance energy transfer (FRET (Sekar and Periasamy, 2003)).

Requirement for high speed, quantum efficiency in live cell imaging

When imaging dynamic or sensitive samples, such as *in vivo* specimens, the benefits of increased spatial and spectral resolution have to be balanced with requirements for fast capture and minimal light exposure to the sample. Dynamic biological events can change on timescales ranging from fractions of a second to minutes or hours. Exposure to light can cause heating or other phototoxic effects that can damage the specimen (Hockberger et al., 1999) and even perturb the biological process of interest. It is therefore essential to optimally utilize the fluorescence signal from the sample. This involves not only detecting the fluorescence photons with high efficiency and low noise but also extracting a maximum amount of information from every detected photon.

Multipoint Confocals

Image capture speed with bright fluorescent signals is largely limited by technical factors, such as detector readout and data acquisition speeds. However, with *in vivo* specimens, signals are typically very low intensity because of the low abundance of fluorophore. High levels of excitation cannot compensate for this limited intensity because of fluorophore saturation (caused by ground-state depletion (Born et al., 1999)) and phototoxicity. These factors, along with the sensitivity of the detectors, present fundamental limits to the speed at which images of sufficient signal-to-noise ratio for analysis can be captured. Weak fluorescent signals are common in many biological applications for reasons ranging from either dim intrinsic fluorescence signals, poor expression of genetic fluorescence markers, non-ideal staining, weak antibody fluorescence labeling or signal attenuation due to scatter and refractive index changes. Traditional laser scanning confocal microscopes (Amos and White, 2003; White et al., 1987) operating with a single aperture and using photomultipliers (PMT) as single pixel readouts are limited in image acquisition rate by the speed of the scanning mirrors and the limited quantum efficiency of the PMT detector. Furthermore, fluorophore saturation provides a fundamental limitation to the achievable signal level. A practical and demonstrated solution involves scanning multiple apertures simultaneously, using a high quantum efficiency multipixel area detector such as a

Charge Coupled Device (CCD) camera. This technique is used in spinning disc confocal microscopes (Fujita et al., 2000), and other multipoint non-confocal microscope designs (Buist et al., 1998; Saggau, 2006; Sinclair et al. 2006). Multipoint confocals have better out-of-focus rejection than slit scanning systems; however cross-talk between adjacent pinholes (which occurs when the light cones from individual pinholes overlap) causes them to have inferior out-of-focus rejection compared to single point confocals. This cross-talk depends on the separation between pinholes. In the swept-field confocal design we use a linear array of 32 pinholes scans the field of view, which is considerably less than that used in a typical spinning disk system. The performance of the swept-field confocal is therefore intermediate between a spinning disk and a single-point confocal system.

A disadvantage of confocal imaging is that signals emanating away from the focal plane are blocked. This often leads to high levels of illumination being used to compensate for a loss of brightness. This in turn can lead to increased levels of photobleaching compared to a bright-field or structured illumination microscope. This is due to the nonlinearity of photobleaching as a function of light intensity. Even though a confocal microscope may deliver the same amount of excitation light as a bright-field microscope on average, the highly concentrated beam at the focus results in a much higher local peak intensity and therefore more bleaching. This added bleaching increases as the confocality increases, with bright-field systems having the least bleaching followed by slit scanners, multipoint confocals, and finally single-point confocals that cause the most bleaching.

Spectral Confocal Microscopy

Spectral resolution in confocal microscopes is often obtained by multiple exposures, using different color optical filters. This technique is easy to implement but can require *a priori* knowledge of the signals but is slow and inefficient in the use of the fluorescent signal. An alternate approach uses a dispersive element (usually a diffraction grating) to spatially separate different wavelength components (Stimson et al., 1999;Hiraoka et al., 2002). The spectrum of a single pixel is collected by a multichannel detector, such as a multi-channel PMT Unfortunately, currently available conventional multi-channel PMTs have low quantum efficiencies (<40%), limited by their photocathodes. The large physical size of the PMT channels requires a high-dispersion grating to separate the spectrum. Simple gratings are inefficient (< 50%) when used with the broadband unpolarized light created by a fluorophore. Prisms are generally more efficient. With the use of custom multilayer coatings, prisms can provide efficiencies well above 95%. However, due to the low dispersion achievable by prisms, it can be challenging to effectively combine a prism with a multichannel PMT. Nevertheless, simple prisms can be used to spread spectra across the limited distance of a few pixels on an Electron Multiplying Charge Coupled Device (EMCCD) area detector to capture the spectra. This is an attractive strategy as EMCCD frame imaging cameras can have very high quantum efficiencies (>90%). Elliot et. al. (Elliott et al., 2012) recently demonstrated a hyperspectral detector based on optical remapping of the hyperspectral image information on a two dimensional detector (Elliott et al., 2012). However, this system requires the collection of hundreds or thousands of frames, making this method slow in practice.

In all of these approaches, the spectral element necessarily changes the optical path, necessitating a complete redesign of the microscope scan head. This often means that hyperspectral scanning and regular monochrome filter based confocal scanning cannot simply be performed on the same instrument.

A spectral element that does not change the optical path can of course always be achieved using multiple optical elements like mirrors or prisms to lead the beam back to the original optic axis after passing the spectral element. An efficient design based on this principle is the prism grating prism configuration (PGP; Aikio, 2001)) used in some commercial line imaging spectrometers (e.g. ImSpector V8E used in (Schultz et al., 2001)) (Herrala et al., 1994; Nagaoka et al., 2012; Owen et al., 2007). This approach requires multiple aligned elements rather than one monolithic bonded part and is therefore typically more complex and space consuming than an Amici prism. Because of the grating used it is still significantly less efficient than a prism (The manufacturer of ImSpector states >50% efficiency for unpolarized light).

Another way of capturing spectrally resolved bright field images on a two dimensional sensor is through different forms of coding such as Hadamard transforms. This technique is, for example employed in (Hanley et al., 2000). The encoding process removes the need for scanning of the sample, but does not reduce the total amount of data that needs to be recorded for a given spectral and spatial resolution. Instead of actuating scanning mirrors the system requires a series of patterns to be applied to the light in the optical path using a spatial light modulator and a series of coded spectral images to be captured from which the hyperspectral image can be reconstructed. Capturing and processing an individual image takes about 30 minutes. Assuming the system was not limited by hardware and processing speed and only by the fundamental issue of fluorophore saturation, the collection could be faster than scanning system with speed, bleaching, and depth and spatial resolution comparable to a bright field microscope. The system does have no built in confocality. Instead Hanley et. al use point spread function deconvolution to achieve some amount of out-of-focus rejection. This technique can be applied to any bright field or confocal microscope and could somewhat improve the point-spread function on any system beyond the system's inherent, hardware-based point spread function.

Our Approach: Prism-based Hyperspectral Multipoint Confocal

For the purposes of imaging dynamic biological events we sought to develop a novel system that combines the benefits of multipoint confocal, prism-based spectral discrimination, and EMCCD detection all in one modular and flexible platform. We present a prism-based approach for adaptation of multipoint confocal microscope designs for fast spectral capture. This approach combines the comparably low loss of prism-based systems along with the speed, signal-to-noise and viability advantages of multipoint confocal scanning. In addition, our method has a key practical implementation advantage in that it allows easy switching from a regular confocal to a hyperspectral system.

We present a design of a spectral imaging system based on a multipoint confocal microscope using an EMCCD camera with a modified double Amici prism (Hagen and Tkaczyk, 2011) as the dispersive element. Amici prisms provide wavelength separation without deflecting the mean direction of the beam and can therefore be easily added to existing systems. They are a low cost solution commonly employed in hand-held spectrometers. By scanning several pinholes in parallel and collecting light with an EMCCD camera, our spectral multipoint confocal system achieves high capture speeds of up to 4 frames per second with 15 channels of spectral resolution. The high quantum efficiency of the EMCCD and the low loss of the prism provide high sensitivity and low noise. We also demonstrate that the addition of hyperspectral resolution does not affect overall image quality and signal-to-noise ratio significantly.

Methods

The Opterra scan head

We modified the scan head of an Opterra multipoint swept field confocal (SFC) microscope (Bembenek et al., 2007; Castellano-Munoz et al., 2012) (Bruker Nano Surfaces FM, Middleton, WI) combined with an EMCCD readout camera (tested with Photometrics QuantEM and Photometrics Evolve Delta, Photometrics, Tucson, AZ). Laser excitation sources at 405 nm, 488 nm, 561 nm, and 640 nm were incorporated in our prototype. The beam path inside the scan head is shown in Figure 1. The excitation light is directed through a set of 32 pinholes and focused on the sample via a dichroic mirror (Chroma zT405/488/561/640rdc or Chroma z488rdc). A series of disconnected points in the sample is illuminated, and the excited fluorescence is imaged through a second set of apertures onto the sensor. An emission filter can be inserted to remove unwanted excitation light from the image. The scan head has a galvanometer actuated mirror to scan the illuminated points horizontally, and a set of piezo scanning mirrors to direct the beam vertically. In normal operation the system sweeps an entire line across the sensor with the galvanometer, exposing a series of non-contiguous lines on the sensor (Figure 1d). After exposure, piezo mirrors shift the beams vertically by one row and horizontal scanning with the galvanometer is continued. This process is repeated 32 times until a complete continuous plane in the sample has been exposed. The image can then be read off the EMCCD camera or the exposure process can be repeated to accumulate more light. The image creation process is outlined in Figure 1.

The Amici prism

We reversibly modified the scanhead by inserting a four component double Amici prism for separation into spectral components into the beam path. Three component Amici prisms have originally been used for the investigation of stellar spectra and today find application as compact wavelength discriminators, such as handheld spectrometers. They are inexpensive and provide wavelength separation without changing the mean beam direction and with minimal loss. We add a fourth component to the prism design to improve the linearity of the spectral separation. The light path inside the prism is shown in Figure 2. The prism consists of four wedges of different materials with different refractive indices and dispersion characteristics bonded together with UV cured optical cement. The entrance and exit faces are coated with a broadband multilayer anti-reflection coating. A simpler design of an amici prism consists of just two materials and translates the optic axis upon transmission. To distinguish from this configuration, the three prism configuration our system is based on is often referred to as a double amici prism. We use the term Amici prism for brevity but always refer to our modification of the more advanced three component design by Amici. This prism is designed to pass a central wavelength of 561 nm without deflection while other wavelengths are deflected as indicated in Figure 2 with linear dispersion over the range 450-700nm (Figure 2c). We measured the overall transmission of the prism to be 94% at 632 nm. The loss is mostly due to residual reflectance and scattering in the interfaces between different wedges, as well as the entrance and exit interfaces. The Amici prism can be moved in and out of the beam path using a plunger.

Image Capture and Reconstruction

After readout, the piezo actuators adjust the field of view horizontally to image an adjacent set of lines in the sample. A second set of spectra is read out as a spectral image S_{ij}. This process is repeated until a continuous region of the sample has been imaged. The result is a frame stack s_{ijk}. The data are then processed to obtain a stack M_{abl} of images of 15 spectral channels by rearranging the elements according to

$$a = fix\left(\frac{j}{15}\right) + k$$

$$b = j$$

$$l = mod(j, 15)$$

This simple treatment assumes that the separation between different pinholes is exactly 15 pixels throughout the sensor area. If this assumption does not apply, aliasing effects need to be taken into account if sub pixel accuracy is desired. The false color rendering of a reconstructed image of fluorescent beads, along with the 15 individual color channels is shown in Figure 3.

System Dichroics and Emission Filters

We implemented two dichroic/filter configurations to prevent light from the excitation laser from being detected by the EMCCD: (1) A dichroic filter directing the excitation laser at 488 nm (Chroma z488rdc) to the sample with a filter bandwidth of ±10 nm and passing all other light between 400 and 800 nm back to the EMCCD. This dichroic is combined with a 500 nm long pass emission filter (Chroma ET500LP) placed directly in front of the EMCCD camera. In this configuration the system uses only the excitation laser at 488 nm and can capture light from 500 nm to 800 nm. (2) A dichroic mirror (Chroma zT405/488/561/640rdc) designed for excitation at 405 nm, 488 nm, 561 nm, and 640 nm, again with a width of about ±10 nm for each wavelength. This dichroic also acts to block the four excitation wavelengths at the camera. In this configuration all four excitation lasers can be used in parallel and light across the entire sensitivity range of the instrument from about 400 nm to about 800 nm can be collected. At the excitation wavelength, no light is detected. Custom dichroic designs can be used to minimize the spectral width of the components blocked in this process. The current width of 20 nm creates clearly visible dips in the collected spectra. The spectral width of the lasers used is less than 5 nm. Custom designed dichroics may reduce the bandwidth of the blocked light to about 5-10 nm, which is less than the width of one channel in the microscope. Rather than being completely blocked, the affected channels then detect a reduced amount of light, which can be compensated in post processing at the expense of additional noise in those channels.

Ghosting Correction

Each pinhole has assigned to it 15 pixel rows on the EMCCD sensor corresponding to 15 color channels. At about 17 nm per channel, this corresponds to a spectral bandwidth of about 255 nm. By selecting which 15 sensor rows are assigned to which pinhole, it is possible to computationally adjust which 250 nm band within the sensitivity range of the microscope should be imaged. If the light emitted by a sample exceeds the 250 nm bandwidth, we observe *ghosting* in which the red light of a certain pinhole falls onto the same pixel row as the blue light of an adjacent pinhole. This effect is only apparent in the images when bright spectral signatures at the extremes of the spectral range of the instrument are encountered. It can in some situations be avoided by computational selection of appropriate wavelengths bins or by choice of appropriate excitation lasers and fluorophores in a sample. It could also be removed by a 250 nm wide band pass filter placed in front of the EMCCD. In practice we used a filter with a pass band from 450 nm to 650 nm (Semrock, Rochester, NY) since it is available as off the shelf item. In addition we are investigating approaches to computationally remove this effect in certain situations. None of the specimens presented in this work required this filter.

Oversampling

To obtain a continuous image, the scanner piezos need to advance the field of view by, at most, the equivalent of one camera pixel between captures. Since the images of different pinholes are approximately 15 pixels apart, 15 images have to be taken. We have the option to adjust the step in scope piezo voltage along with the amount of captured images to allow us to zoom the image in one dimension. For example, we can capture 30 frames for one image, moving the field of view by the equivalent of about ½ pixel between captures. The resulting image then contains 15 channels with a spatial resolution of 512 by 1024 pixels. In the images presented here a slight oversampling was chosen so that 18 frames make up one image.

Wavelength calibration

The beam path through the scan head has been simulated with Zemax software (Radiant Zemax, Redmond, WA) and wavelength separation is designed to be approximately linear. Due to the complexity of the setup and the unpredictable effects of potential small alignments in any of the components, it is necessary to calibrate the system using a known source after insertion of the Amici prism. We evaluated three primary methods for calibration.

One method was to image different specimen of CdSe quantum dots (Lumidot, Sigma-Aldrich, St. Louis, MO) with emission wavelengths of 520 nm, 560 nm, 590 nm, 610 nm, and 640 nm. Quantum Dots (QD) provide a reasonably precise and accurate spectral reference, that is easy to prepare and can be excited with a single wavelength (Cutler et al., 2013). We computed the sum over all columns of the sensor

$$F = \sum_i S_{ij}$$

for each of the images. We then identified the spectral peaks with a multi-step peak finding algorithm:

- Generate a peak list by finding all maxima in F.
- For each peak F_p in the list calculate the local average $A = \sum_{p=p_0-10}^{p_0+10} \frac{F_p}{20}$and remove the peak from the list if $F_p < 5A$.
- For each remaining peak compute a score quantifying how many other peaks are approximately 15 pixels away from the peak based on the equation $G = \sum_i mod\left(F_p - F_i, 15\right)$
- Retain the 32 peaks with the highest score.

This algorithm makes it possible to find peaks reliably even in noisy and distorted data. For a given QD image with a given wavelength, the extracted peaks mark the rows on the sensor where light of the corresponding wavelength is collected. We repeat the process for all available QD samples and create an array λ_p assigning a wavelength to the detected rows. The rows that are half way between a 520 nm row and the adjacent 640 nm row are marked as boundaries between the areas assigned to different pinholes. Finally we interpolate linearly to assign wavelengths to the rows in between the peaks.

With this information we created a 512 by 512 pixel map assigning a pinhole and wavelength to every pixel of the readout sensor. This method is used to study the linearity of the prism in the wavelength range in question.

Additionally, we capture calibration data by placing a diffuser on the sample holder of the microscope and illuminating it with 50 mW laser diodes of different wavelengths. A red diode at 662 nm, a frequency doubled 532 nm green diode, and a 406 nm blue diode. The fundamental wavelengths and other peaks we attributed to parasitic nonlinear processes are clearly visible in the diode spectra and can be used in the calibration process. Compared to the QD samples, the laser diodes provide brighter and narrower spectra. The number of spectral lines that fall within the wavelength range of the microscope (three or four), is however, too small to provide a good calibration, especially without a good model of the system distortion to fit to the collected data points. This diode calibration method was used to verify the QD calibration.

The third calibration method uses the excitation lasers of the microscope rather than external sources. For calibration, a diffuser is placed on the sample holder and the emission filter is removed. Individual frames are captured with each of the excitation lasers illuminating the diffuser. These images are then used for calibration in a similar way as in the above methods. This method is the easiest to implement on the system and was used before each sample capture to verify calibration.

Pinhole Selection

Three different pinhole sizes are evaluated by imaging the spectrum of a red laser diode. The diode illuminates a diffuser placed on the specimen holder of the microscope. The spectrum of the laser measured by the CCD is shown in Figure 5 for different pinholes. The results indicate that the 45 micron pinhole offers the best tradeoff between resolution and signal strength for a 40x lens (NA 1.3).

Color Rendering

To render images in false color, we generate a transform matrix that converts a 15-element vector into a 3 element RGB pixel value. For a true color representation the matrix is composed of the 15 element absorption spectra of the three kinds of cones in the human eye. A simpler transform is a 3x15 identity matrix mapping channels 1 to 5 to red, channels 6 to 10 to green and channels 11 to 15 to blue. False color images computed with an identity matrix along with selected individual spectra are shown in Figure 3. For false color renderings in all other images, the mapping is adjusted to map the channels that contain the respective fluorophore contributions to the red, green and blue color channels.

Linear Unmixing

If there is spectral overlap between multiple markers in a sample, those markers will appear together in several channels. In the presence of this crosstalk, signal processing techniques are required to isolate the signals of individual fluorophores. Many approaches are available for the unmixing of spectral information, often transferred from fields that also commonly encounter hyperspectral data (Harris, 2006). One approach applied in gamma ray spectroscopy and analytical chemistry is known as spectral stripping and allows the removal of strong unwanted known spectra by subtracting them from the collected data (Busch et al., 1974).

A hyperspectral image containing only the emission of known fluorophores can be expressed as a superposition of a set of known fluorophore spectra. If the constituent spectra of an image are arranged as the rows of a matrix M and the relative weights of the spectra present in a pixel are described in a vector P, the resulting spectrum S of that pixel can be found by

$$S = M\,P$$

To compute P from a measured spectrum S, we use the inverse of M:

$$P = M^{-1}S$$

By solving this equation we identify which spectra are present in a pixel. This technique is known as linear unmixing (Zimmermann, 2005; Zimmermann et al., 2003) and is widely employed in spectral analysis of microscopy data. Advanced techniques include improved methods for inversion of the matrix M, such as maximum likelihood analysis (Davis and Shen, 2007; Neher and Neher, 2004). Since this method requires a priori knowledge of the dye spectra, it is however not suited for samples with unknown fluorescent emitters. More importantly, small variations in the fluorescence spectrum as they frequently occur in response to such factors as different chemical environments, developmental fluorophore stage, and refractive effects in the sample can decrease the performance of the methods. The methods will also fail if the system is not sufficiently calibrated and detected spectra no longer match the reference spectra in the library (for example because of the effects of any excitation blocking filters). This is a common problem of unmixing methods that is difficult to diagnose and rectify. It is especially challenging in the analysis of live samples where the sample preparation is time consuming, has significant potential for errors, and a large expected variation in different samples.

A cytofluorogram (Demandolx and Davoust, 1997) is a simple method that allows the separate small numbers of unknown dye spectra. This method has also been used to handle changes in dye spectra due to chemical environment.

The problem of separating a number of unknown spectra in an image can be described as a matrix multiplication

$$I = L \cdot N$$

Where in our case of 15 spectral channels, I is a 15 by p matrix of the spectra of all the p pixels in the image, L is a 15 by n matrix of the n constituent spectra that make up each pixel in the data, and N is a n by p matrix of the relative strength of the constituents in each of the image pixels. The task is to find a factorization LN that adequately describes the collected data I.

We evaluated two different analysis methods to perform blind spectral separation: Principal Component Analysis (Jolliffe, 2002) (PCA) and Nonnegative Matrix Factorization (NNMF) (Berry et al., 2006). Both are available as MATLAB (MathWorks, Natick, MA) software functions. PCA has previously been applied to spectral separation in microscopy data (Zimmermann, 2005). Our analysis shows that it is not an ideal method because it allows for the extracted spectral intensities to be negative. NNMF performs a constrained optimization of the matrixes L and N to ensure that they are both positive. Approaches similar to NNMF, such as Multivariate Curve Resolution, have been applied to microscopy data of high spectral resolution (de Juan et al., 2008). We find that for low noise spectral data of limited complexity, NNMF is easy to implement using MATLAB and performs well for the spectral analysis of our test samples. More involved methods are a subject of further research and beyond the scope of this publication that focuses of the capabilities of the capturing hardware, performing spectral separation only as a means of assessing the system performance.

Biological Sample Methods

Fixed transgenic Zebrafish

To demonstrate spectral imaging in biologically relevant specimen we use a zebrafish larva from the *flk1*:GFP vascular endothelial transgenic line (Thompson et al., 1998) stained for glial fibrillary acidic protein (GFAP). The vasculature is labeled with GFP, GFAP is labeled with Alexa 568 (red) and cell bodies are labeled with DAPI (Vector Laboratories, Burlingame, CA). We also use a second fixed zebrafish larva from the *sox9b*:EGFP transgenic line (Plavicki et al., 2014) in which collagen type II is labeled with Alexa 568 (red). The specimen is mounted in Vectashield with DAPI (blue) labeling the cell nuclei.

Cell mitosis in a live Xenopus laevis embryo

X. laevis embryos were prepared as described in (Woolner et al., 2008). Briefly, *X. laevis* embryos were microinjected at the two-cell stage with 0.08ng eGFP-a tubulin mRNA, 0.02ng mCherry-Histone H2B mRNA, and 0.31ng mTagBFP-CAAX mRNA in 5 nL water per cell. All mRNA was transcribed using mMESSAGE mMACHINE SP6 Transcription Kit (Ambion) from pCS2 backbones. Embryos were cultured at 17ºC in 0.1x MMR and imaged approximately 22 hours post fertilization.

Immune response in the inner ear of a live zebrafish larva

Double transgenic zebrafish larvae *Tg (mpeg1:dendra2)* crossed with the transgenic line *Tg(mpx:mCherry)* were used as previously reported(Harvie et al., 2013). The transgenic fish were microinjected in the otic vesicle at 72 hours post fertilization (hpf) with ~300 colony forming units (CFU) of *Streptococcus iniae* (Fuller et al., 2001). Bacteria were labeled with 5 μM CellTracker Blue CMAC dye (Molecular Probes, Invitrogen, Grand Island, NY) according to the manufacturer’s instructions. Briefly, bacteria were pelleted by centrifugation at 1500 x *g* for 5 min and resuspended in a 5 µM solution of CellTracker Blue dye in 1 ml phosphate-buffered saline (PBS) and incubated for 30 min at 37°C. Bacteria were pelleted by centrifugation and allowed to recover for 30 min at 37°C in Todd-Hewitt medium (Sigma-Aldrich, St. Louis, MO) supplemented with 0.2% yeast extract (BD Biosciences, San Jose, CA) and 2% proteose peptone (vegetable) (Sigma-Aldrich, St. Louis, MO). Blue-labeled *S. iniae* were prepared for microinjection as previously described (Harvie et al., 2013). For bacterial infection and live imaging, larvae were anesthetized in E3 medium containing 0.2 mg/ml tricaine (ethyl 3-amino-benzoate; Sigma-Aldrich, St. Louis, MO). Anesthetized larvae were settled onto the bottom of a custom-made, glass-bottom dish. To prevent pigment formation, larvae were maintained in E3 containing 0.2 mM N-phenylthiourea (Sigma-Aldrich, St. Louis, MO).

Results

Capture Speed

The fastest technically possible capture speed is determined by the highest possible frame rate of the readout camera. Since one spectral image is composed of at least 15 frames, the fastest speed is 15 times slower than the frame rate of the readout camera, which is 66 frames per second for the Evolve 512 Delta EMCCD (Photometrics, Tucson, AZ) and 31.5 frames per second for the QuantEM:512 SC (Photometrics, Tucson, AZ), resulting in a spectral frame rate of about 4 and 2 frames per second respectively.

Noise Comparison

The signal-to-noise ratio in an image can generally be attributed to three different sources. (1) Readout noise: all noise introduced by the readout of the image from the EMCCD. This noise depends on the number of readouts and is independent of exposure time or the strength of the detected signal. This noise is small if the EM gain of the camera is used to boost signal and could be eliminated completely with the use of photon counting. (2) Background noise: noise generated by the detector dark current and by stray ambient light. The amount of background noise is dependent on the exposure time. (3) Amplification (e.g. electron multiplication) noise: noise dependent on the gain provided by the sensor. This could, in principle, be eliminated entirely by using photon counting.

Adding spectral resolution capabilities to the microscope will influence the signal-to-noise ratio in captured images in two ways: (1) Losses introduced by the prism will weaken the signal and scattering in the prism may provide background noise. (2) Readout noise from the EMCCD is due to the fact that at least 15 readouts are required to capture the data for one image.

To evaluate the loss introduced by the spectral separation, we imaged a FluoCells prepared slide (Life Technologies, Carlsbad, CA) of muntjac skin fibroblast cells labeled with three different stains. Filamentous actin in these cells was labeled with Alexa Fluor 488 phalloidin, mitochondria were labeled with Alexa Fluor 555 and nuclei were labeled with the far-red fluorescent TO-PRO-3 nucleic acid stain. The slide was imaged with four different configurations (Figure 7): (i) A non-spectral single frame exposure (i.e. no Amici prism in path) with multipoint confocal microscope. This image represents the optimally achievable signal quality (Figure 7a). (ii) The complete spectral system (i.e., Amici prism in path) using a total of 18 frames to compose one image (Figure 7d). (iii)The data captured with the spectral system, but showing the sum of all color channels to obtain a monochrome image comparable to the optimal image (i), but containing the noise of the spectral image (ii) (Figure 7c). (iv) Finally, a frame average of 18 frames without the spectral system in place. This image contains noise due to readout, but does not contain the noise due to loss and scattering in the prism (Figure 7b). In all cases, a four band Sedat emission filter with notches at 405/488/561/640m (Chroma, Bellow Falls, VT) was used to block emission lasers. The settings in all cases are otherwise identical with the exposure time chosen such that the aggregate exposure times are identical in all cases. The difference between Figure 7a and 7b is caused by the noise resulting from additional readouts, with the use of EMCCD gain this difference is barely noticeable. The difference between 7b and 7c, due to the loss introduced by the Amici prism and any image distortions in image reconstruction is practically imperceptible.

To illustrate the capacity of our system to separate similar spectra, a test specimen of Focal Check Green-Green double stained fluorescent beads (Life Technologies, Carlsbad, CA) is shown in Figure 8 along with the result of a three-component nonnegative matrix factorization. The beads are 6 micron in diameter and contain a dye with a fluorescence peak at 525nm in their core and a dye peaking at 512 nm in their shell (Ducros et al., 2009; Zucker et al., 2007). The spectra were found and separated without significant crosstalk by the algorithm in the regions where the spectra were not saturated (see Figure 8). Saturation is present in the center of the image and prevents the separation.

Biological Specimen Testing

A number of fixed and live specimens were imaged to show the spectral sensitivity of the device for imaging biological samples. Figure 9 shows an image of a specimen of fixed muntjac cells labeled with three different fluorophores (Life Technologies, Carlsbad, CA) captured with a 60x oil 1.4 NA objective (Nikon, Mehlville, NY). This figure illustrates clear separation of the three labels. Filamentous actin is labeled with Alexa Fluor 488 Phalloidin, mitochondria are labeled with Alexa Fluor 555 and nuclei are labeled with the far-red fluorescent TO-PRO-3 nucleic acid stain. Further, a screen capture of the "live" image collection of this specimen while moving the slide on the stage demonstrates the capture speed. The video was recorded in real time with a capture speed of 2 frames per second and is included as supplementary movie 1.

In addition to the commercially prepared muntjac slide, additional samples were examined to have more variety in fluorophores for spectral system performance evaluated. Figure 10 shows an image of our fixed sample of a zebrafish larva. The vasculature is shown in green and cell bodies in blue. The second zebrafish larva is shown in Figure 11. The red stain shows the developing fin rays in the pectoral fin. As with the muntjac slide, both zebrafish samples showed good spectral fidelity with the expected fluorophore spectra.

As the strength of the hyperspectral multipoint system described is the ability to collect spectral images at high speed, two live samples are demonstrated. Cell mitosis in a live *Xenopus laevis* embryo is shown in supplementary movie 2. This movie was captured with a 500 ms exposure every 5 seconds. A single hyperspectral frame at 500 ms exposure time is shown in Figure 12. The cell membrane was labeled with targeted mTagBFP and shown in blue, microtubules were labeled with eGFP-atubulin displayed in green, and chromatin is visualized with mCherry-Histone H2B.

Supplementary movies 3 and 4 and Figure 13 show the immune response in the inner ear of a live zebrafish larva captured with a 40x water immersion lens. Bacteria were labeled with Cell Tracker Blue CMAC (Molecular Probes) and injected into the inner ear. Macrophages were labeled with Dendra and neutrophils with mCherry. The macrophages appear yellow due to photoconversion of the Dendra fluorescent protein by the blue excitation laser. Bacteria tend to form aggregates, but blue points representing small aggregates or possibly individual cells are also visible. Strong excitation is necessary to excite the weak blue dye. This leads to visible green autofluorescence in some structures. This autofluorescence is spectrally different from Dendra and does not photoconvert. We found that both red-labeled neutrophils and green-labeled macrophages were recruited to the site of infection in the inner ear and began phagocytosing blue-labeled bacteria. Macrophages that had phagocytosed bacteria slowed in motility as they became rounded in appearance. Engulfed bacteria could be seen within vesicles inside the macrophages.

Supplementary movie 3 shows the specimen imaged with 250 ms exposure time, capturing one frame every 10 seconds. Neutrophils are moving slowly or are inactive; while the movement of macrophages is the most prominent feature at this time scale. Small bacteria aggregates are visible, but their motion is too fast to track individual signals from frame to frame. In this capture configuration, the fast capture speed helps to resolve features by reducing motion blur.

Supplementary movie 4 was captured from the same specimen at maximum instrument speed of 4 frames per second. At this high speed the dynamics of small aggregates or individual bacteria is visible. The macrophages also show fast dynamics that can only be visualized at this frame rate.

Discussion

Our Amici prism dispersion element is easily integrated into an existing multipoint confocal design, such as the Opterra multipoint confocal (Bruker Nano Surfaces FM, Middleton, WI). It adds spectral resolution while only moderately impacting other important parameters of the system, most importantly the sensitivity and capture speed. Our analysis (Figure 7) shows that the bulk of added noise is introduced by the additional required detection readouts. The readout noise is dependent on the gain applied to the camera and this difference can be small for high EM amplification factors and could disappear in a photon counting setup. The noise introduced by the Amici prism is hard to detect (if at all) by simple visual analysis of the images. We conclude that in most practical cases, the losses due to spectral capture are so small that there is no need for regular (non-spectral) confocal capture. It appears advantageous to always capture hyperspectral data even if they are not immediately required for a particular experiment as the penalty for doing so is negligible. Traditional monochrome images can still be created from the hyperspectral data if desired, but the available additional spectral information can still be accessed if needed. The only scenario that still requires traditional monochrome or filter based capture is if frame rates exceeding 4 full frames per second are required.

Spectral PSF

Apart from the technical limitation of the achievable pixel size, the spectral resolution of our system is limited by the optical point spread function (PSF) of the Amici prism spectrometer. The width of this PSF is expected to be proportional to the size of the pinhole in use. The spectra of a monochromatic source imaged using different pinholes (shown in Figure 5) indicate that the point-spread function of our system is smaller than a pixel of our current camera. If the spread of the peaks shown in the figure was due to the system PSF, the peak would have to be twice as wide for a 60 micron pinhole as it is for a 30 micron pinhole. The widths are however practically identical. We attribute the detected spread to electron leakage between adjacent EMCCD pixels.

Once the camera resolution approaches the spectral PSF, spectral resolution could be further increased by insertion of a longer beam path for the Amici prism along with a mask with fewer pinholes. This would sacrifice speed to improve spectral resolution. It may also be possible to place two or more Amici prisms in series to allow additional flexibility in the choice of spectral resolution.

Prism spectrometers are typically nonlinear in their mapping of frequency to separation distance. The Amici prism used was designed to minimize nonlinearities using Zemax simulations. Linearity was also tested with the different calibration methods. The prism is found to be linear to within the current resolution of the instrument and the calibration methods used (17nm per channel) (Fig. 2c).

Spatial PSF

The spatial point spread function is not affected by the insertion of the Amici prism and the reconstruction process. The point spread function in all dimensions on a reconstructed hyperspectral image is shown in Figure 6a. A direct comparison of the PSF of the standard swept field confocal microscope with the PSF of a spectral channel is shown in Figure 6b.

Capture Speed

The capture speed of the Amici multipoint system is directly related to the frame rate of the readout camera. The Photometrics Evolve Delta provides the current state of the art of 66 full frames per second at 512 by 512 pixels resulting in about 4 hyperspectral images per second. In addition to future faster readout cameras, the hyperspectral frame rate could be increased through binning or the use of regions of interest, resulting in an improved frame rate at the expense of spectral or spatial resolution or field of view. It is also possible to undersample the images by taking less than the required 16 frames to compose one image. For example, using 2x2 binning and capturing 8 frames per hyperspectral image would half the spatial and spectral resolution but provide about 16 spectral images per second.

The shorter exposure time improves the effective resolution of the images by reducing motion blur even if the acquisition process does not require a fast continuous capture. This is shown in Figure 12 in a comparison of an image of an oocyte cell during mitosis captured with a 60X oil immersion lens and 500ms exposure time. While the longer exposure time makes the image seem less noisy, details like the dynamic microtubules, are only discernable in the short exposure image due to motion blur.

Spectral Resolution

We achieve a spectral resolution of about 17 nm per channel over 15 channels. The spectral resolution is determined by the pixel pitch of the readout camera. Other camera models like the QImaging Rolera EM (QImaging, Surrey, BC, Canada) have roughly half the pixel pitch of our currently used cameras and would double the spectral resolution to about 30 channels and 8.5 nm per channel. The spectral bandwidth of about 255 nm could be adjusted by designing a different Amici prism. This resolution is sufficient to perform spectral unmixing to discern overlapping fluorophores as shown in Figure 8.

High Speed Spectral Resolution

Hyperspectral images eliminate the need for specialized fluorophore dependent filters, and allow the selection and adjustment of spectral filtering after the capture process. Much of the potential of hyperspectral datasets may be realized through post-capture computational spectral analysis and unmixing. Extending the information extracted from samples that exhibit fast dynamics together with strong spectral overlap of multiple fluorophores is a focus of our future research.

Application to Spinning Disc Multipoint Confocals

Spinning disc multipoint confocals use a rapidly spinning disc with separated pinholes to scan the sample rather than scanning an array of pinholes using mirrors. The pinholes are arranged such that after one rotation of the disc the entire sample has been scanned and a complete continuous image has been projected onto the readout camera. The camera typically captures over multiple rotations of the disc until enough light has been collected. It is conceivable that a spinning disc system using an amici prism could be devised, possibly with the help of further post processing, but it is not immediately clear what would be a practical way to do this. Simply placing an amici prism in front of the readout camera may work as long as it is possible to read out multiple images during one rotation of the disc to prevent the spectra of different sets of pinholes from overlapping in the images. One could also use 15 different discs with different constant pinhole separations, though it is hard to imagine such a system being practical.

Conclusion

We present a multipoint confocal microscope with 15 channel spectral resolution captured efficiently, without degrading the speed and sensitivity of the system. Spectral and spatial data are orthogonal sources of information. Increasing spectral resolution therefore significantly increases the information content obtained from microscope images. Putting this information to optimal use is, however, not trivial. The establishment of new techniques in microscopy is driven by availability and wide acceptance by the biomedical community. We believe that the facility of hyperspectral imaging in light microscopy will enable users to explore this rich information source and find applications beyond current uses, for example in the adoption of new intrinsic and extrinsic fluorescent labels.

Acknowledgements

This research was supported by National Institutes of Health grants R44MH065724 and R01GM102924, and additional funding from the UW-Madison Graduate School and Morgridge Institute for Research. The Bement Lab was funded by NIH grant GM052932. Matthew Larson was funded in part by the UW Medical Scientist Training Program. We wish to thank Jeff Stuckey, Dale Osterlie, Dan Daugherty, John Ritz and Michael Szulczewski of Bruker Nano Surfaces FM for useful testing, scanning firmware modifications and advice. We acknowledge the efforts of Julie Last, Trillian Gregg and Brian Burkel of LOCI for system use and project input. We also wish to thank Photometrics Inc. for assistance with cameras and the Bement and Heideman laboratories for specimens for imaging.

References

Aikio, M., (2001) PhD thesis, Department of Electrical Engineering, University of OuLu, Finland.

Amos, W.B., and White, J.G. (2003). How the Confocal Laser Scanning Microscope entered Biological Research. Biol. Cell *95*, 335–342.

Bembenek, J.N., Richie, C.T., Squirrell, J.M., Campbell, J.M., Eliceiri, K.W., Poteryaev, D., Spang, A., Golden, A., and White, J.G. (2007). Cortical granule exocytosis in C. elegans is regulated by cell cycle components including separase. Development. *134*, 3837–3848.

Berry, M.W., Browne, M., Langville, A.N., Pauca, V.P., and Plemmons, R.J. (2006). Algorithms and applications for approximate nonnegative matrix factorization. In Computational Statistics and Data Analysis, pp. 155–173.

Bertani, F.R., Ferrari, L., Mussi, V., Botti, E., Costanzo, A., and Selci, S. (2013). Living matter observations with a novel hyperspectral supercontinuum confocal microscope for VIS to near-IR reflectance spectroscopy. Sensors *13*, 14523–14542.

Bini, J., Spain, J., Nehal, K., Hazelwood, V., DiMarzio, C., and Rajadhyaksha, M. (2011). Confocal mosaicing microscopy of human skin ex vivo: spectral analysis for digital staining to simulate histology-like appearance. J. Biomed. Opt. *17*.

Born, M., Wolf, E., Bhatia, A.B., Clemmow, P.C., Gabor, D., Stokes, A.R., Taylor, A.M., Wayman, P.A., and Wilcock, W.L. (1999). Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light (Cambridge ; New York: Cambridge University Press).

Buist, Müller, Squier, and Brakenhoff (1998). Real time two-photon absorption microscopy using multi point excitation. J. Microsc. *192*, 217–226.

Busch, K.W., Howell, N.G., and Morrison, G.H. (1974). Elimination of interferences in flame spectrometry using spectral stripping. Anal. Chem. *46*, 2074–2079.

Carlsson, K., Danielsson, P.E., Lenz, R., Liljeborg, A., Majlöf, L., and \AAslund, N. (1985). Three-dimensional microscopy using a confocal laser scanning microscope. Opt Lett *10*, 53–55.

Castellano-Munoz, M., Peng, A.W., Salles, F.T., and Ricci, A.J. (2012). Swept field laser confocal microscopy for enhanced spatial and temporal resolution in live-cell imaging. *18*, 753–760.

Cutler, P.J., Malik, M.D., Liu, S., Byars, J.M., Lidke, D.S., and Lidke, K.A. (2013). Multi-color quantum dot tracking using a high-speed hyperspectral line-scanning microscope. PLoS ONE *8*.

Davis, L.M., and Shen, G. (2007). Extension of multidimensional microscopy to ultrasensitive applications with maximum-likelihood analysis. In Proc. SPIE, p. 64430N–64430N–12.

Demandolx, D., and Davoust, J. (1997). Multicolour analysis and local image correlation in confocal microscopy. J. Microsc. *185*, 21–36.

Denk, W., Strickler, J.H., and Webb, W.W. (1990). Two-photon laser scanning fluorescence microscopy. Science *248*, 73–76.

Dickinson, M.E., Bearman, G., Tille, S., Lansford, R., and Fraser, S.E. (2001). Multi-spectral imaging and linear unmixing add a whole new dimension to laser scanning fluorescence microscopy. Biotechniques *31*, 1274.

Ducros, M., Moreaux, L., Bradley, J., Tiret, P., Griesbeck, O., and Charpak, S. (2009). Spectral Unmixing: Analysis of Performance in the Olfactory Bulb <italic>In Vivo</italic>. PLoS ONE *4*, e4418.

Elliott, A.D., Gao, L., Ustione, A., Bedard, N., Kester, R., Piston, D.W., and Tkaczyk, T.S. (2012). Real-time hyperspectral fluorescence imaging of pancreatic Î^2-cell dynamics with the image mapping spectrometer. J Cell Sci *125*, 4833–4840.

Fujita, K., Nakamura, O., Kaneko, T., Oyamada, M., Takamatsu, T., and Kawata, S. (2000). Confocal multipoint multiphoton excitation microscope with microlens and pinhole arrays. Opt. Commun. *174*, 7 – 12.

Fuller, J.D., Bast, D.J., Nizet, V., Low, D.E., and de Azavedo, J.C. (2001). Streptococcus iniae virulence is associated with a distinct genetic profile. Infect. Immun. *69*, 1994–2000.

Greinberger, C., and Konnerth, A. (2012). Imaging calcium in neurons. Neuron. *73*, 862–885.

Hagen, N., and Tkaczyk, T.S. (2011). Compound prism design principles, I. Appl Opt *50*, 4998–5011.

Hanley, Q.S., Verveer, P.J., Arndt-Jovin, D.J., and Jovin, T.M. (2000). Three-dimensional spectral imaging by hadamard transform spectroscopy in a programmable array microscope. J. Microsc. *197*, 5–14.

Harris, A.T. (2006). Spectral mapping tools from the earth sciences applied to spectral microscopy data. Cytom. A *69*, 872–879.

Harvie, E.A., Green, J.M., Neely, M.N., and Huttenlocher, A. (2013). Innate immune response to Streptococcus iniae infection in zebrafish larvae. Infect. Immun. *81*, 110–121.

Helmchen, F., and Denk, W. (2005). Deep tissue two-photon microscopy. Nat Meth *2*, 932–940.

Herrala, E., Okkonen, J.T., Hyvarinen, T.S., Aikio, M., and Lammasniemi, J. (1994). Imaging spectrometer for process industry applications. pp. 33–40.

Hiraoka, Y., Shimi, T., and Haraguchi, T. (2002). Multispectral imaging fluorescence microscopy for living cells. Cell Struct Funct *27*, 367–374.

Hockberger, P.E., Skimina, T.A., Centoze, V.E., Lavin, C., Chu, S., Dadras, S., Reddy, J.K., and White, J.G. (1999). Activation of flavin-containing oxidases underlies light-induced production of H2O2 in mammalian cells. Proc. Natl. Acad. Sci. U.S.A. *96*, 6255–6260.

Jolliffe, I.T. (2002). Principal Component Analysis (Springer).

De Juan, A., Maeder, M., Hancewicz, T., and Tauler, R. (2008). Use of local rank-based spatial information for resolution of spectroscopic images. J. Chemom. *22*, 291–298.

Lam, P.Y., Waterman, C.M., Fischer, R.S., and Huttenlocher, A. (2014). Spinning disk confocal imaging of neutrophil migration in zebrafish. Methods Mol. Biol. *1124*, 219–233.

Lavis, L.D., and Raines, R.T. (2008). Bright ideas for chemical biology. ACS Chem. Biol. *3*, 142–155.

Lerner, J.M., Gat, M., and Wachman, E. Approaches to spectral imaging hardware. In Curr. Proc. Cytom., p. Chapter 12:Unit 12.20.

Nagaoka, T., Nakamura, A., Okutani, H., Kiyohara, Y., and Sota, T. (2012). A possible melanoma discrimination index based on hyperspectral data: a pilot study. Skin Res. Technol. *18*, 301–310.

Neher, R., and Neher, E. (2004). Optimizing imaging parameters for the separation of multiple labels in a fluorescence image. J. Microsc.-Oxf. *213*, 46 – 62.

Owen, D.M., Auksorius, E., Manning, H.B., Talbot, C.B., de Beule, P.A.A., Dunsby, C., Neil, M.A.A., and French, P.M.W. (2007). Excitation-resolved hyperspectral fluorescence lifetime imaging using a UV-extended supercontinuum source. Opt. Lett. *32*, 3408–3410.

Paddock, S.W. (1999). Confocal laser scanning microscopy. Biotechniques *27*, 992–1007.

Petran, M., Hadravsky, M., Benes, J., and Boyde, A. (1986). In Vivo Microscopy Using the Tandem Scanning Microscopea. Ann. N. Y. Acad. Sci. *483*, 440–447.

Pinaud, F., Michalet, X., Bentolila, L.A., Tsay, J.M., Doose, S., Li, J.J., Iyer, G., and Weiss, S. (2006). Advances in fluorescence imaging with quantum dot bio-probes. Biomaterials *27*, 1679–1687.

Plavicki, J.S., Baker, T.R., Burns, F., Hofsteen, P., Xiong, K., Gooding, A., Peterson, R.E., and Heideman, W. (2014). Characterization of sox9b transgenic line: a tool for studying development and toxicology. Under Review.

Rizzo, M.A., Davidson, M.W., and Piston, D.W. (2009). Fluorescent protein tracking and detection: fluorescent protein structure and color variants. In Cold Sptring Harb. Protoc., (Cold Spring Harbor, NY),.

Saggau, P. (2006). New methods and uses for fast optical scanning. Curr. Opin. Neurobiol. *16*, 543 – 550.

Schultz, R.A., Nielsen, T., Zavaleta, J.R., Ruch, R., Wyatt, R., and Garner, H.R. (2001). Hyperspectral imaging: A novel approach for microscopic analysis. Cytometry *43*, 239–247.

Sekar, R.B., and Periasamy, A. (2003). Fluorescence resonance energy transfer (FRET) microscopy imaging of live cell protein localizations. J. Cell Biol. *160*, 629–633.

Sheppard, C., Shotton, D., and Sheppard, C. (1997). Confocal laser scanning microscopy (BIOS Scientific Publishers Oxford).

Thompson, M.A., Ransom, D.G., Pratt, S.J., MacLennan, H., Kieran, M.W., Detrich, H.W., 3rd, Vail, B., Huber, T.L., Paw, B., Brownlie, A.J., et al. (1998). The cloche and spadetail genes differentially affect hematopoiesis and vasculogenesis. Dev. Biol. *197*, 248–269.

White, J.G., Amos, W.B., and Fordham, M. (1987). An evaluation of confocal versus conventional imaging of biological structures by fluorescence light microscopy. J. Cell Biol. *105*, 41–48.

Woolner, S., O'Brien, L.L., Wiese, C., and Bement, W.M. (2008). Myosin-10 and actin filaments are essential for mitotic spindle function. J. Cell Biol. *182*, 77–88.

Zimmermann, T. (2005). Spectral Imaging and Linear Unmixing in Light Microscopy. Adv Biochem EnginBiotechnol *95*, 245–265.

Zimmermann, T., Rietdorf, J., and Pepperkok, R. (2003). Spectral imaging and its applications in live cell microscopy. FEBS Lett. *546*, 87 – 92.

Zipfel, W.R., Williams, R.M., and Webb, W.W. (2003). Nonlinear magic: multiphoton microscopy in the biosciences. Nat Biotech *21*, 1369–1377.

Zucker, R.M., Rigby, P., Clements, I., Salmon, W., and Chua, M. (2007). Reliability of confocal microscopy spectral imaging systems: use of multispectral beads. Cytom. A *71*, 174–189.

Supporting Information

Figures and Captions

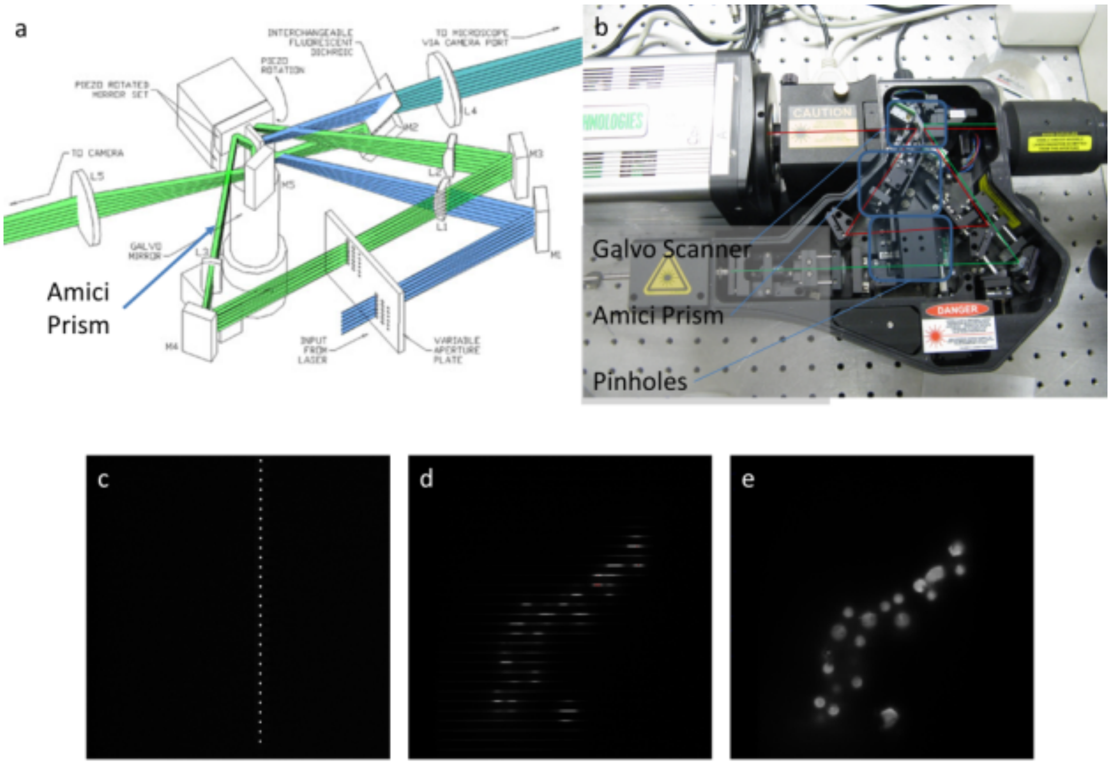


Figure 1: Hardware setup and monochrome image acquisition. (a) Schematic of the Opterra multipoint confocal scan head. (b) Photograph of the scan head with the prism. The prism is mounted on a plunger and can be moved in and out of the beam path. (c) Image captured by the readout camera when the beam scanners are inactive. (d) Image captured by the readout camera while scanning only with the horizontal (galvanometer) scanners. (e) Image captured by the readout camera while scanning both the horizontal and the vertical (piezo) scanners.

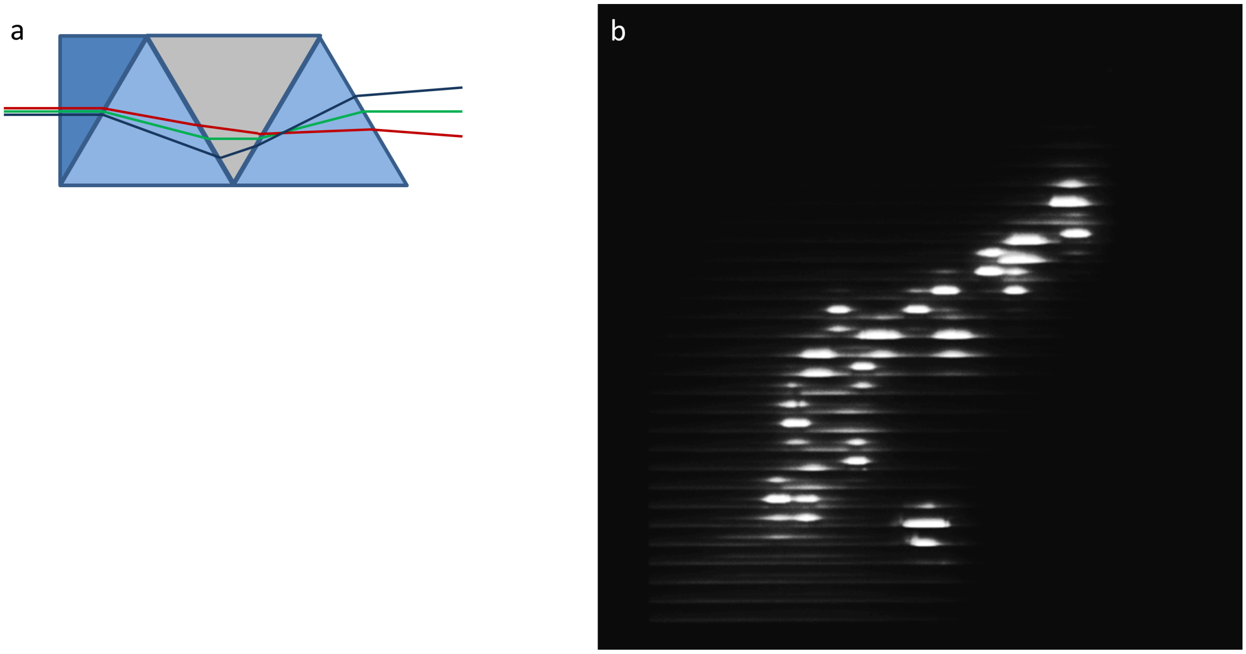


Figure 2: Wavelength separation with the Amici prism. (a) The Amici prism is a compound prism that separates different wavelengths without changing the beam direction. (b) When the Amici prism is in the beam path, an image exposed while operating only the horizontal (galvanometer) scanners contains the spectra of 32 lines in the sample.

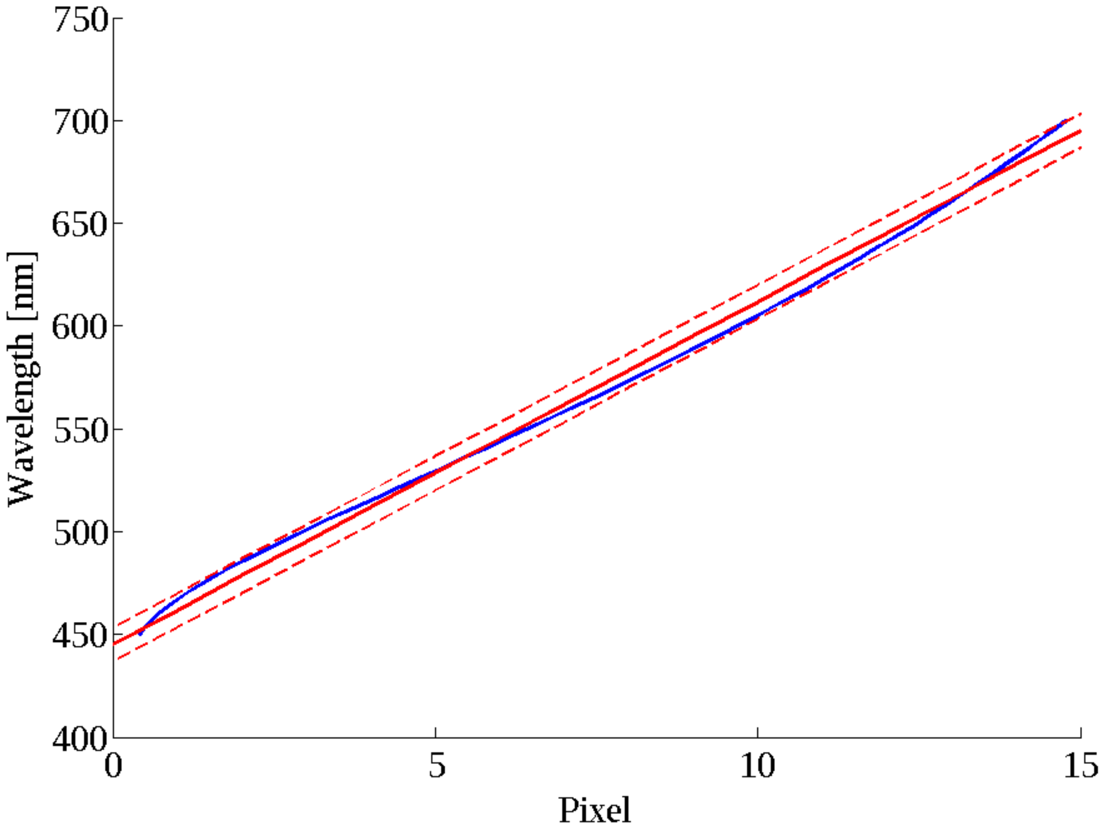


Figure 2c: Spectral separation of the Amici prism on the detector simulated with ZMAX (blue line). The pixel pitch is 16 micron. The red line indicates an ideal linear separation and the dashed red lines delineate an area of $\pm\frac{1}{2}$ pixel or spectral channel around the ideal. The Amici prim is thus linear within the spectral resolution of our microscope.

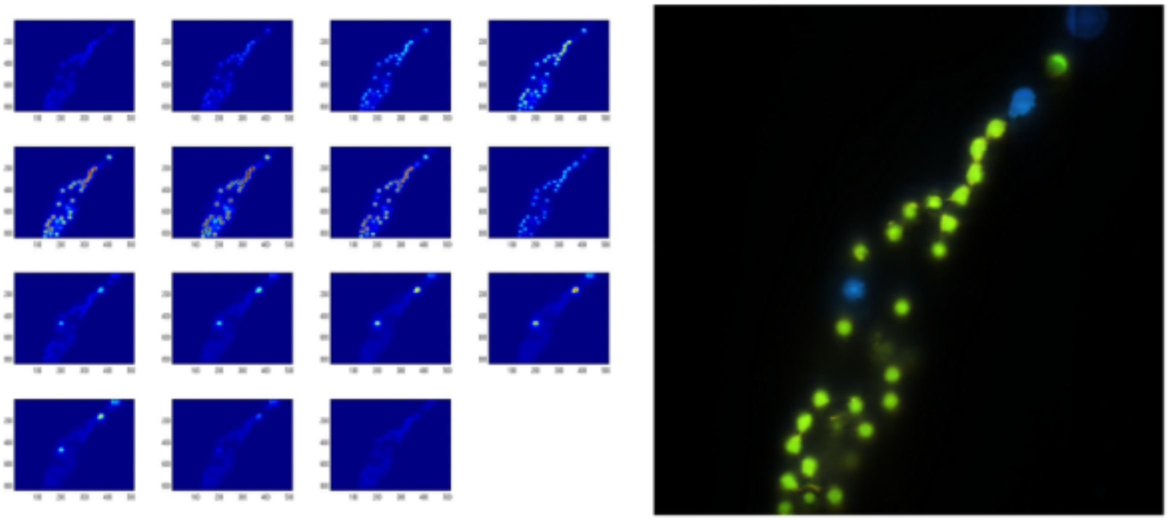


Figure 3: The 15 Spectral channels of a sample of fluorescent beads (Molecular Probes F-21010 and F-8842), as well as a false color rendering. Captured with filter configuration 1 (see section "System Dichroics and Emission Filters").

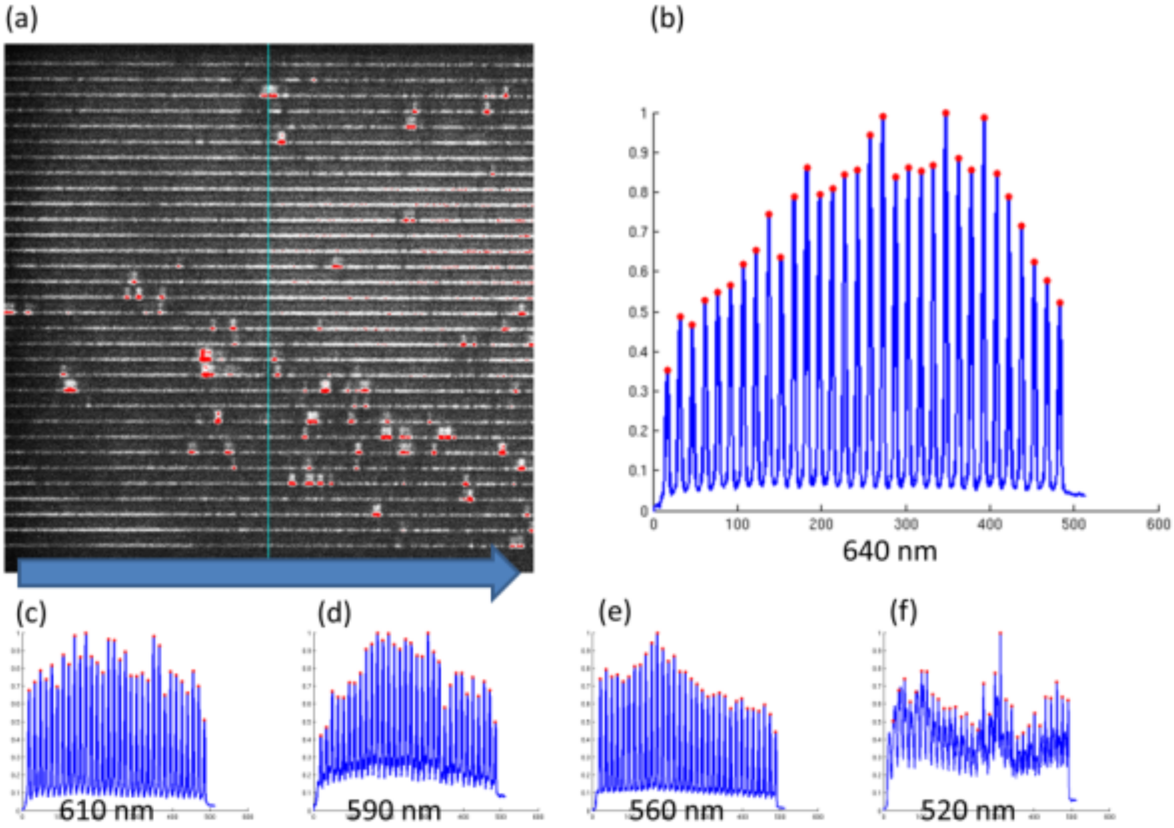


Figure 4: Calibration Data. To calibrate and test the microscope we image a set of quantum dot samples that emit at different wavelengths. Subfigure (a) shows a single frame taken of a QD sample with an emission peak at 640 nm. Subfigure (b) shows the sum of all columns in of (a) The positions of the 32 peaks in this plot, are the positions on the sensor where 640 nm light from the 32 pinholes is detected. Figures (c-f) show similar plots for samples with different known emission peaks.

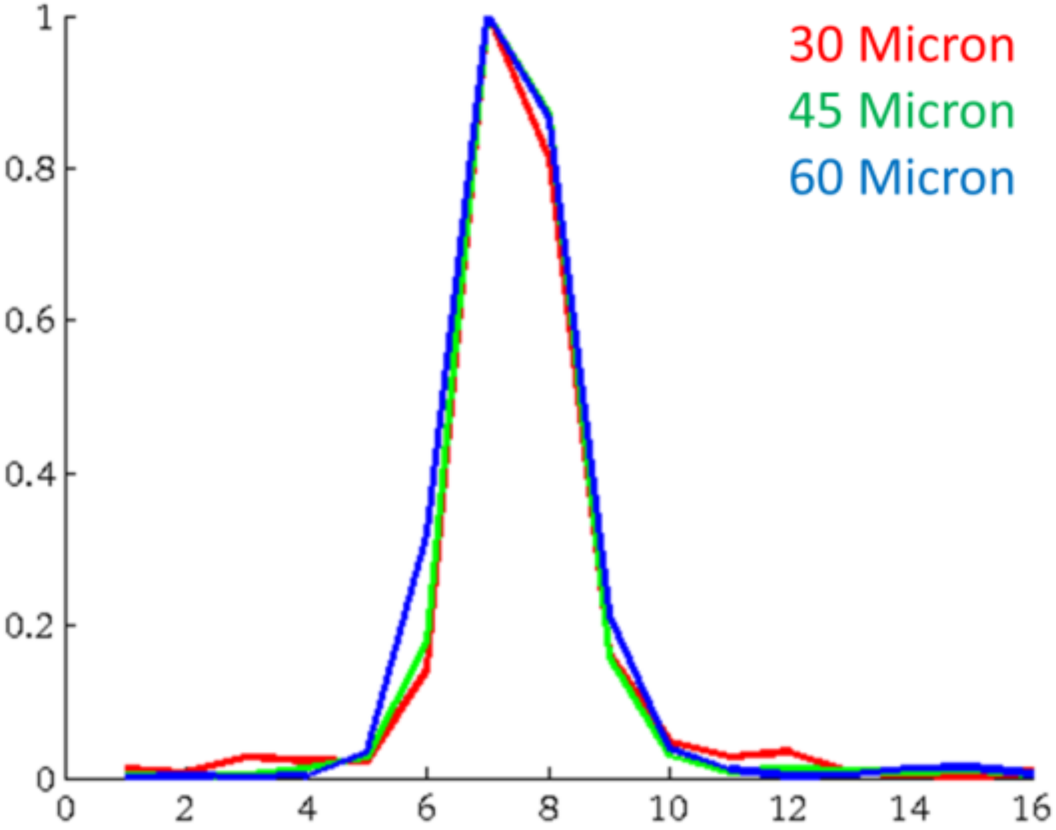


Figure 5: Spectral point spread functions. The point spread functions for different size apertures are measured by shining a laser diode on a diffuser placed in the image plane of the microscope objective.

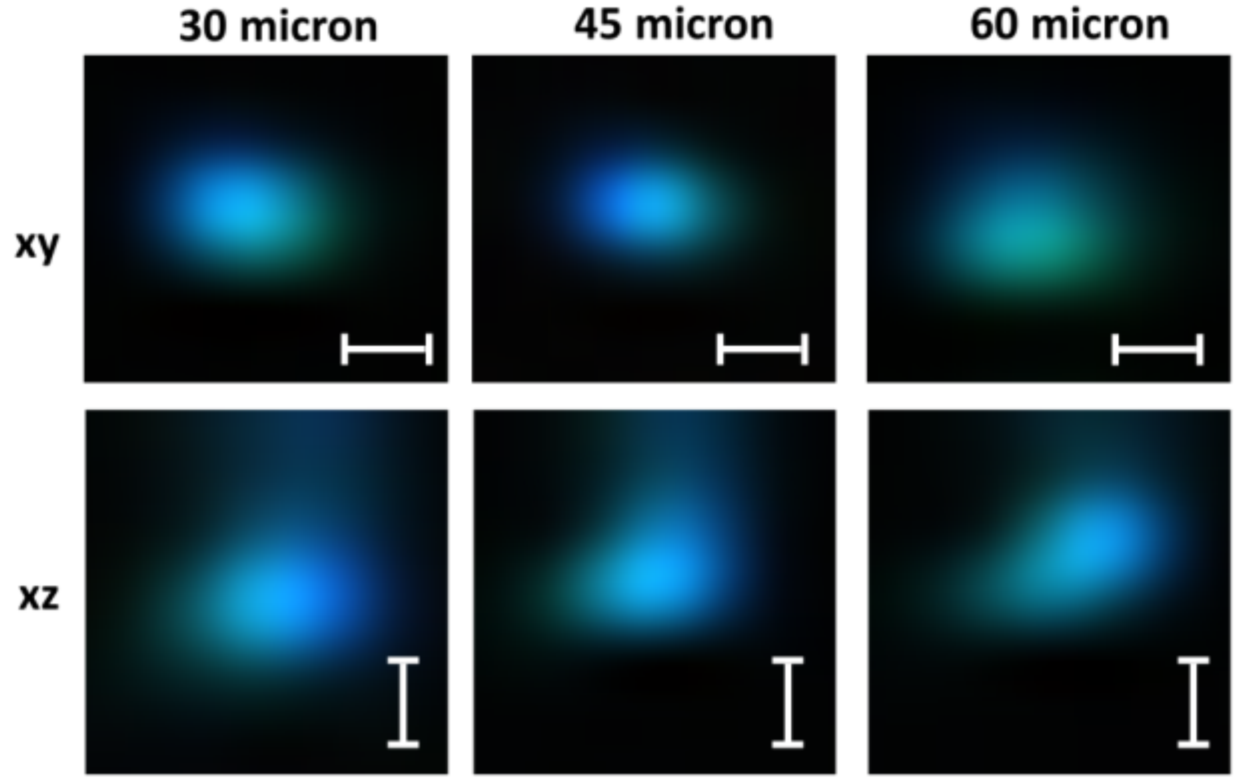


Figure 6a: Spatial point spread functions for different pinholes. These images are obtained by imaging 0.2 micron large fluorescent beads with a 60x oil objective (NA 1.4). The size marker indicates 1 micron. In all cases the Point Spread Function extends over only a few camera pixels.

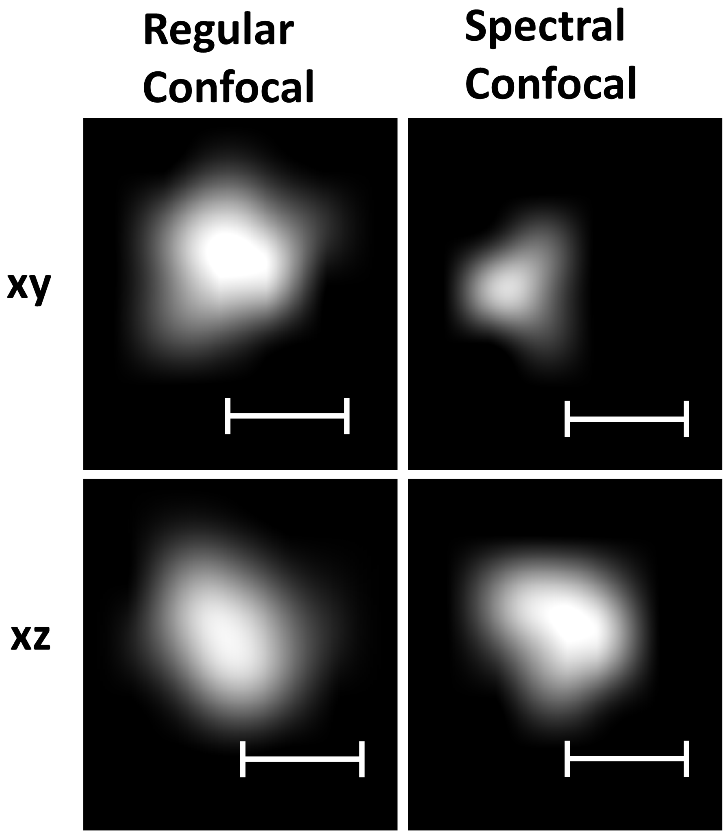


Figure 6b: Spatial point spread functions for regular swept field capture and spectral capture. These images are obtained by imaging 0.2 micron large fluorescent beads with a 40x oil objective. The size marker indicates 1 micron. The spectral separation does not negatively impact the point spread function of the instrument.

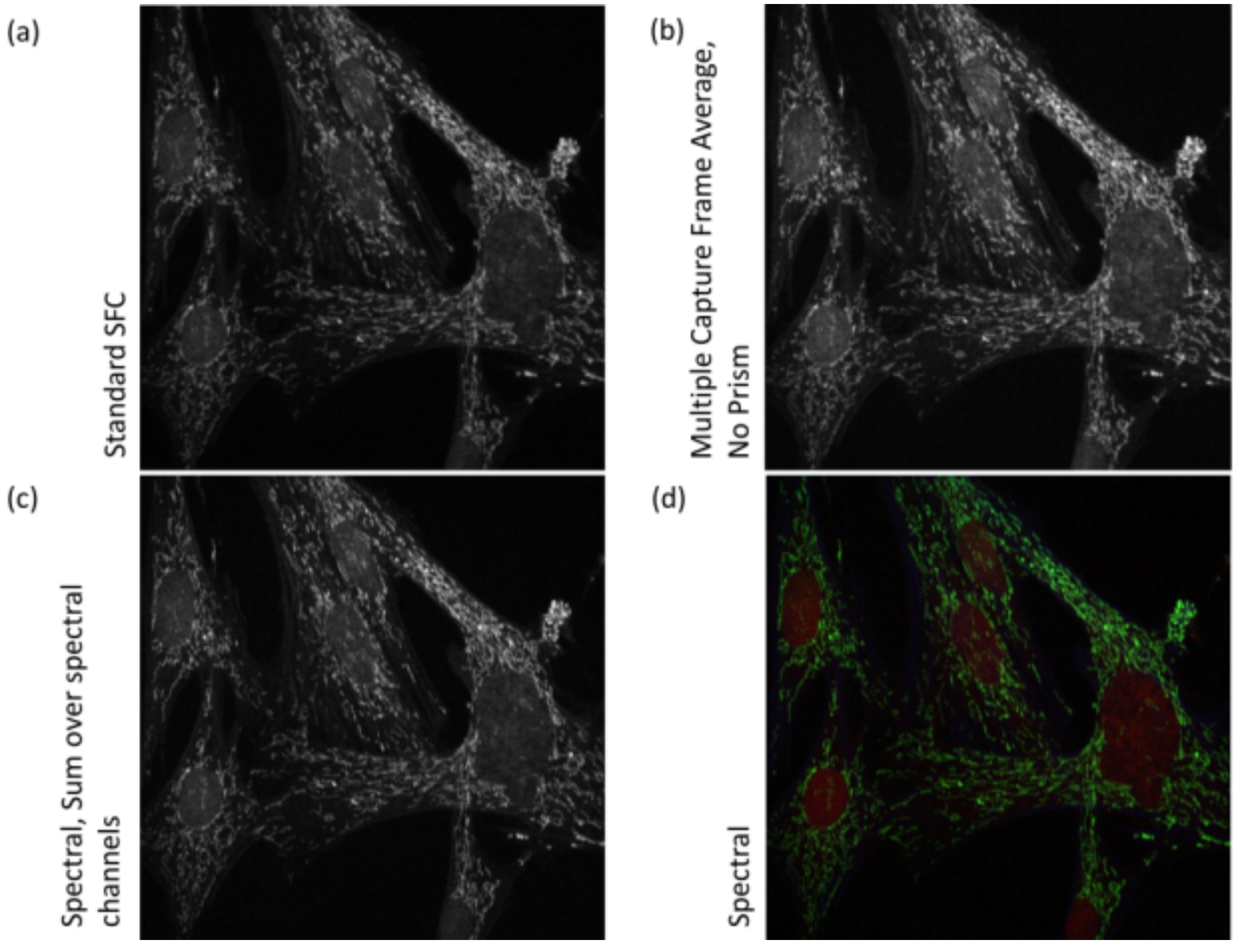

(a)
Standard SFC
(b)
Multiple Capture Frame Average, No Prism
(c)
Spectral, Sum over spectral channels
(d)
Spectral

Figure 7: Noise analysis for low gain: (a) Single exposure image, 600 ms (b) 18 frames, 35ms frame average. (c) Sum of spectral channels 18 frames, 35 ms each. (d) Spectral image, 18 frames 35 ms each. The sample is the muntjac specimen also shown in Figure 9. The image was captured with filter configuration 2 (see section "System Dichroics and Emission Filters").

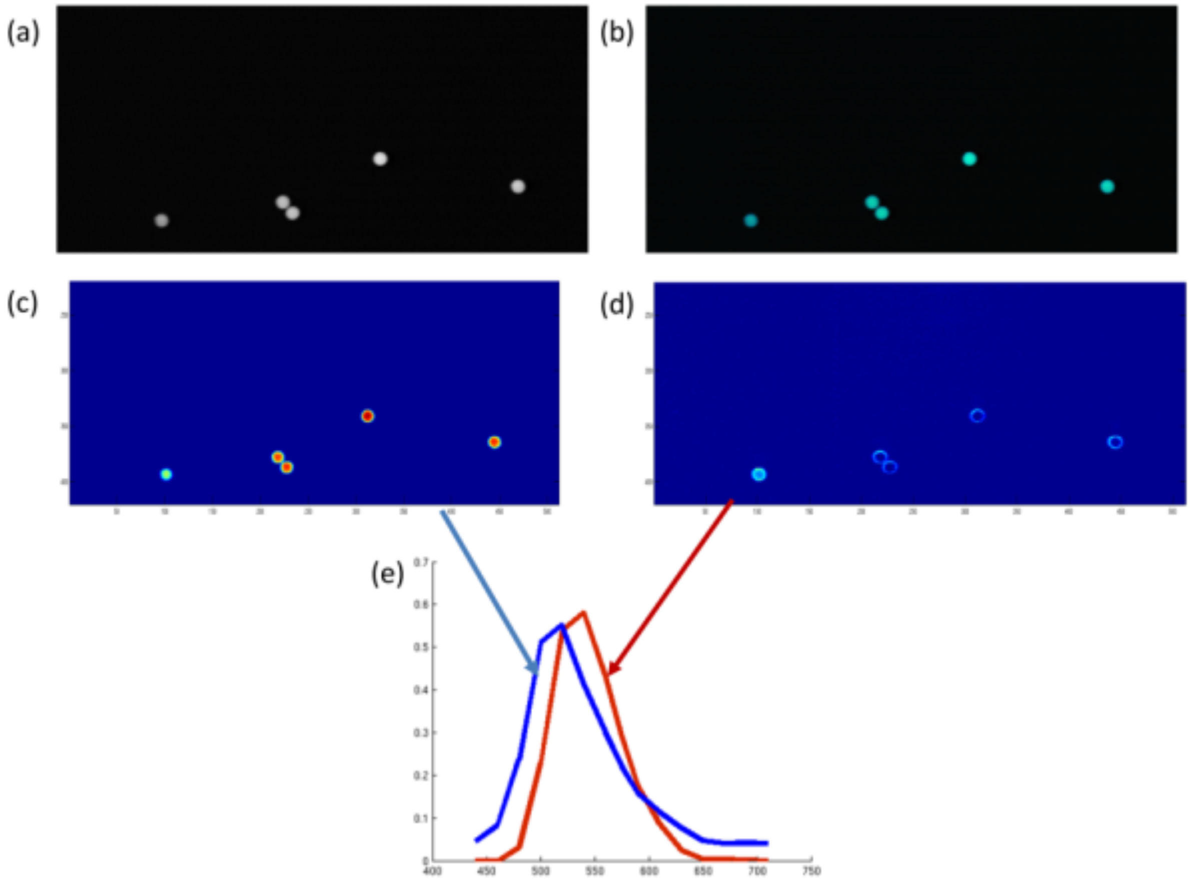


Figure 8: Blind spectral analysis of Invitrogen Green-Green Beads. The beads are 6 micron in diameter and have different dies in their core and shell. (a) Monochrome image of the beads. (b) False color spectral image of the beads. (c) First channel obtained by a NNMF spectral analysis. (d) Second channel (e) The spectra corresponding to the extracted channels. The image was collected with filter configuration 1 (see section "System Dichroics and Emission Filters").

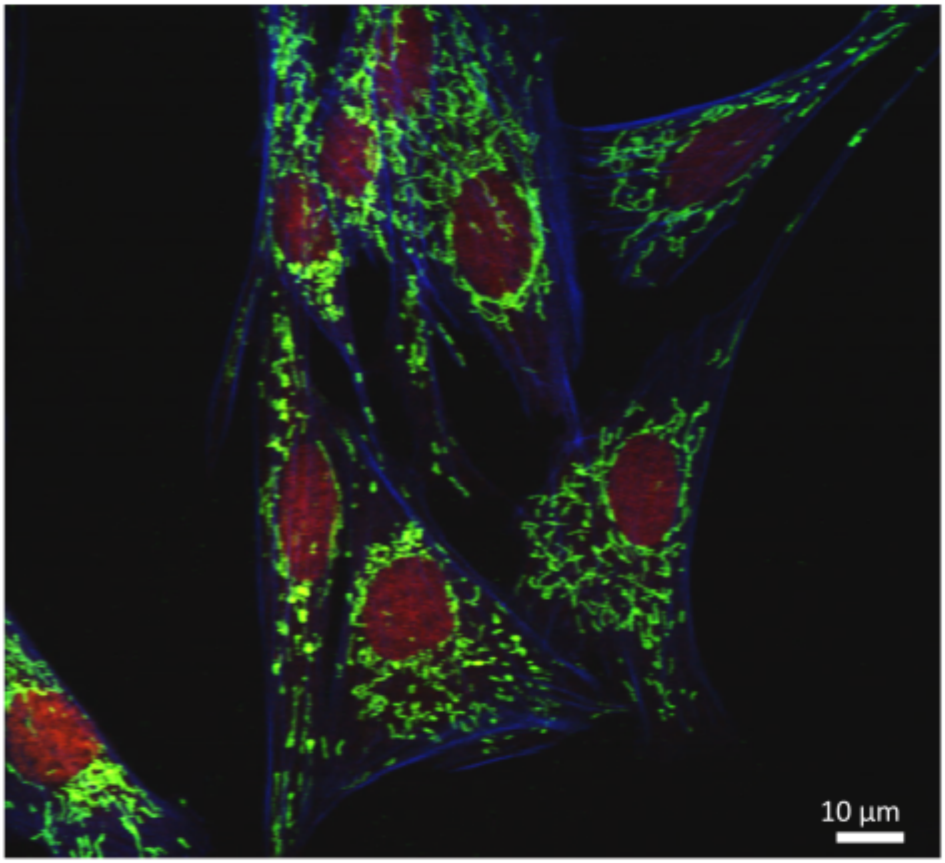


Figure 9: False color rendering of FluoCells commercial prepared slide (Life Technologies, Carlsbad, CA): muntjac. Filamentous actin is labeled with Alexa Fluor 488 Phalloidin, mitochondria are labeled with Alexa Fluor 555 and nuclei are labeled with the far-red fluorescent TO-PRO-3 nucleic acid stain. The image was captured with filter configuration 2 (see section "System Dichroics and Emission Filters") using a 60x oil objective and excitation at 488 nm, 561 nm, and 640 nm.

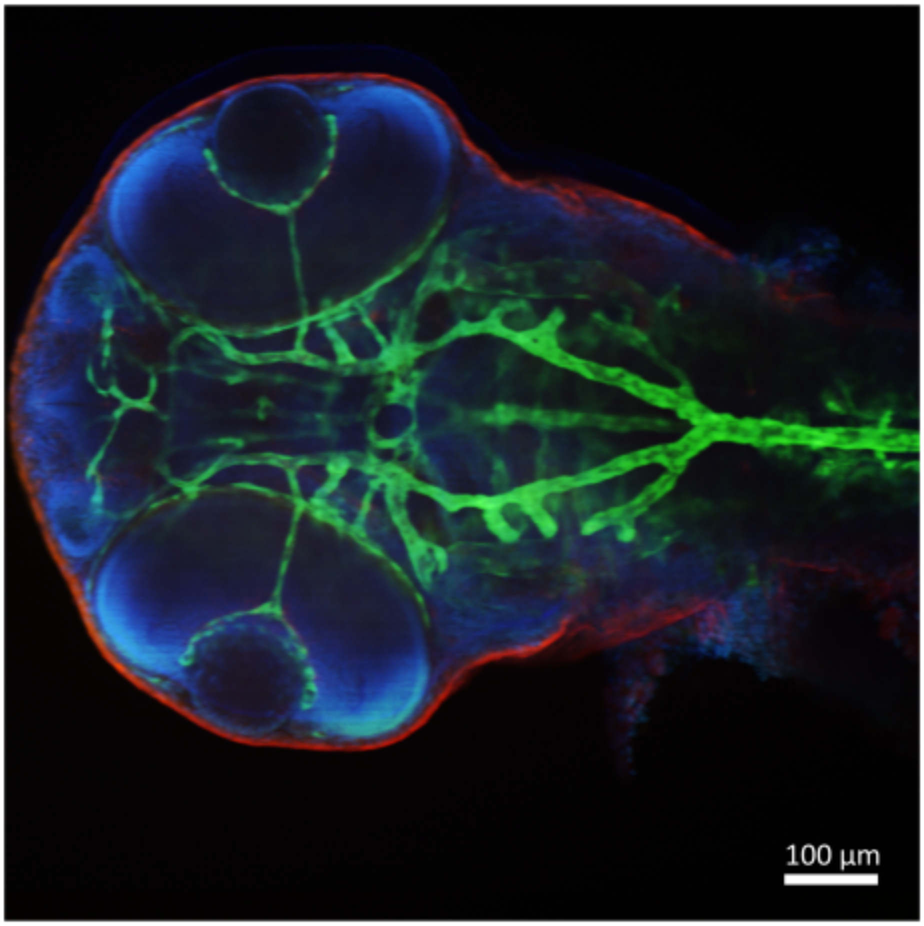


Figure 10: Transgenic zebrafish embryo (flk1:GFP also known as kdrl: GFP) stained for glial fibrillary acidic protein (GFAP) labeled with Alexa 568 and mounted in Vectashield mounting medium containing 4,′6′-diaminido-2-phenylindole (DAPI) (Vector Laboratories, Burlingame, CA). Flk is an early marker for endothelial cell progenitors and this image shows the developing vasculature. The image was captured with filter configuration 2 (see section "System Dichroics and Emission Filters") using a 10x air objective and excitation at 405 nm, 488 nm, 561 nm, and 640 nm.

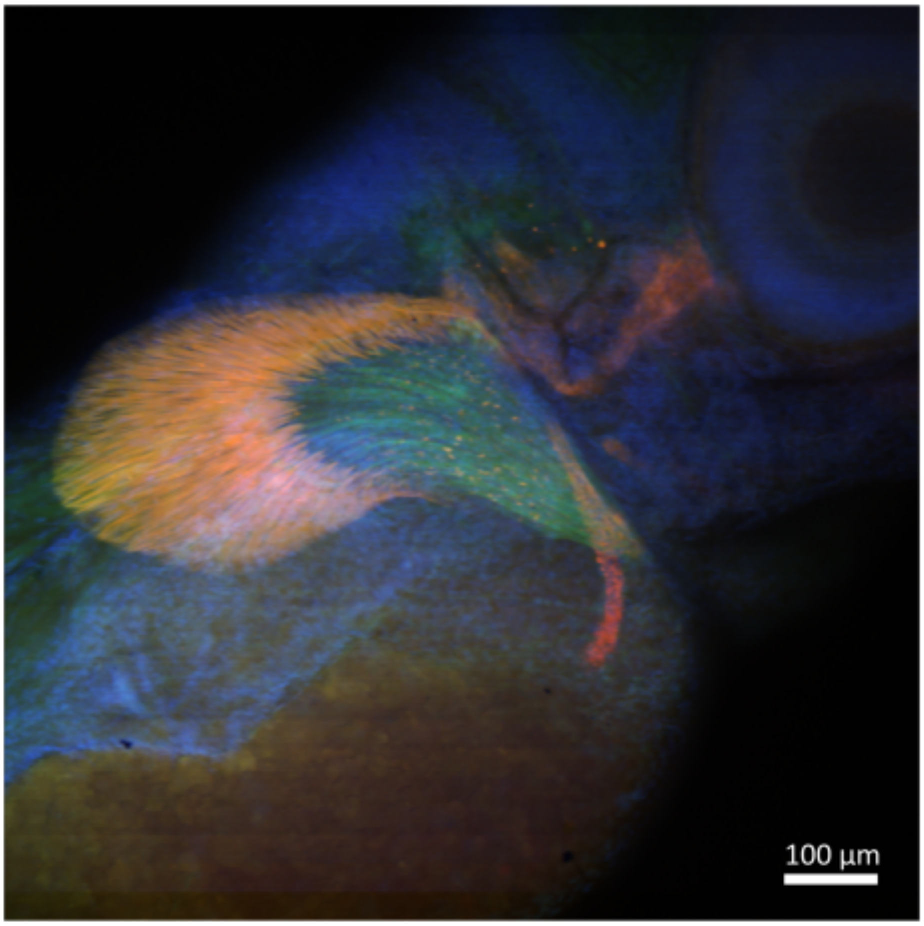


Figure 11: Transgenic zebrafish embryo (sox9b:EGFP) stained for collagen 2a, which is labeled with Alexa 568 (red), and mounted in Vectashield with DAPI (blue). One can see sox9b expression in the developing fin rays in the pectoral fin. The image was captured with filter configuration 2 (see section “System Dichroics and Emission Filters”) with a 10x air objective and excitation at 405 nm, 488 nm, 561 nm and 640 nm.

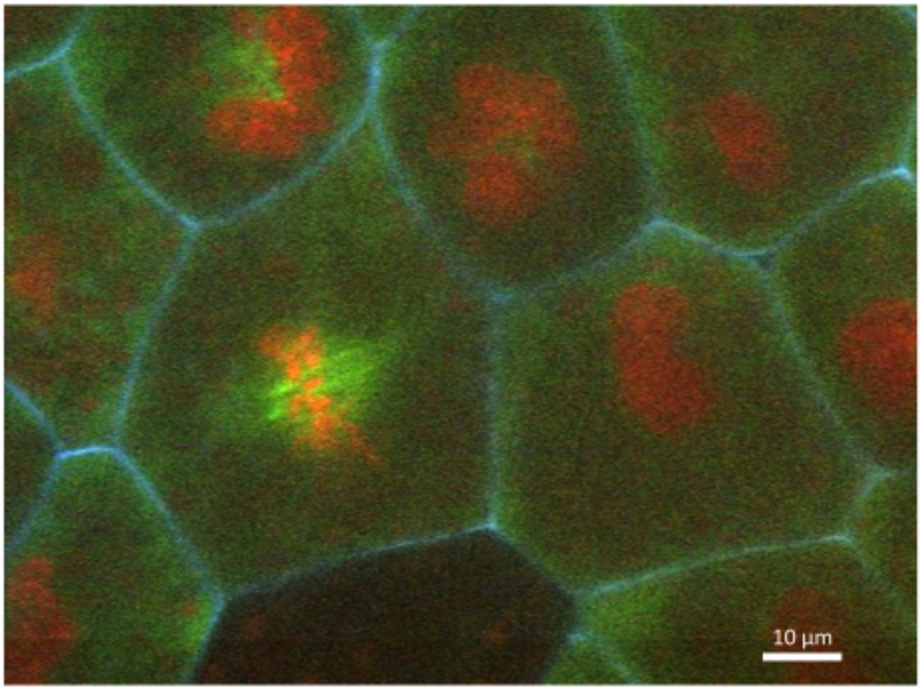


Figure 12: Frame from a video of mitosis in a live *Xenopus laevis* embryo. The cell membrane is labeled with targeted mTagBFP, microtubules are labeled with eGFP-atubulin, and chromatin is visualized with mCherry-Histone H2B. The image was captured with 500 ms exposure time. A video of the mitosis is provided in supplementary movie 2. The image was captured with filter configuration 2 (see section "System Dichroics and Emission Filters") with a 60x oil objective and excitation at 405 nm, 488 nm, and 561 nm.

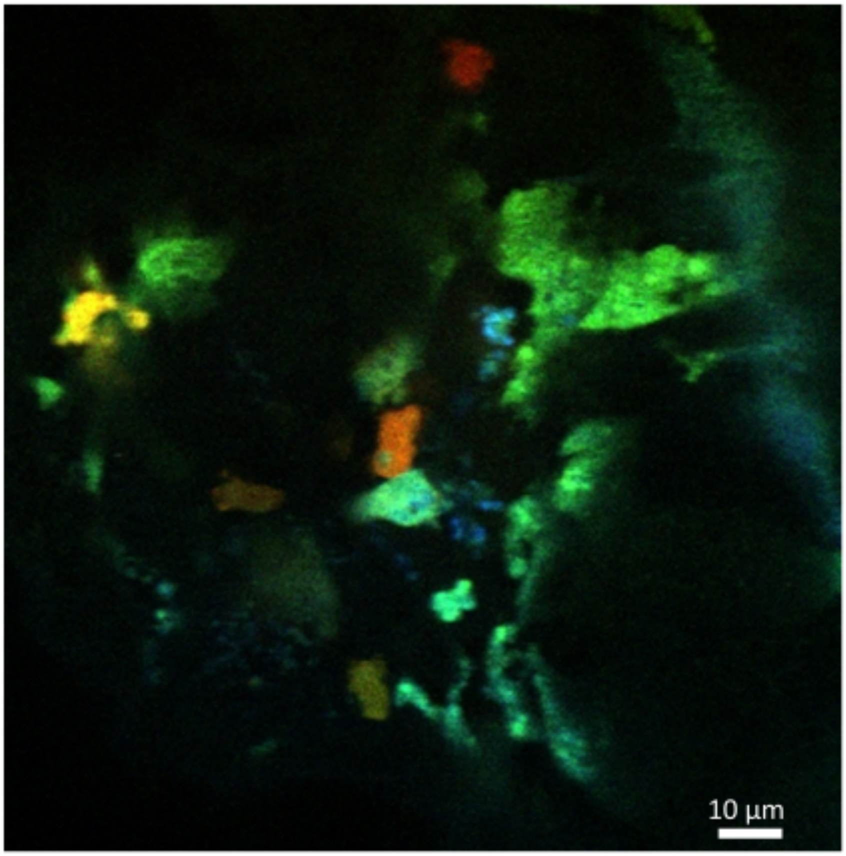


Figure 13: Innate immune response in zebrafish. Bacteria labeled with Cell Tracker Blue CMAC (Molecular Probes, Cat. no. C2110) injected into the inner ear of a live zebrafish larva. Macrophages are labeled with Dendra (green, yellow after photoconversion) and neutrophils with mCherry (red). The view contains one neutrophil on the top center, as well as several macrophages. Small aggregates of bacteria are visible in the center of the image. See also supplementary movies 3 and 4. The image was captured with filter configuration 2 (see section "System Dichroics and Emission Filters") using a 40x water immersion objective and excitation at 405 nm, 488 nm, and 561 nm.

Chapter 8: Future Directions

In this thesis I outlined our work in creating a open source toolkit largely based on ImageJ with some additional open source algorithm work in MatLab for analysis of microscopy data. ImageJ as discussed in chapter 2 has gone beyond a simple application and been extended into a toolkit and a library. ImageJ's strength has always been as a tool for the biological enduser, its ability for bench biologists to not only use the code but easily extend is why ImageJ has had such a long period of active use and development. It is actually this user base of thousands of biologists and not the actual code base that makes ImageJ so valuable and impactful. However like all software packages it is important to take stock of the current applications and recognize that regular major feature addition and sometime even periodic refactoring is necessary to keep pace with the expanding user base and application base it serves. This was especially important with ImageJ as it had its creation in the early 1990s where 2D acquisition was the standard. Now in modern biological microscopy 3D imaging over space and time is the standard often with additional channels of information collection. As we discussed ImageJ needed to be extended to accommodate these additional modes of operation. This resulted in ImageJ2, which is now in regular use and had a interesting approach to development. It would have much easier to start from scratch but in doing so would have alienated the very use base that was most important to retain. So by adding a compatibility layer we were able to retain the user base with support for previous code and features while adding a new date engine and support for new data types. This ImageJ2 has now been in regular distribution and has achieved its primary goal. It is still in use by the biologists, there was no fork or splitting of the community between a ImageJ1 or ImageJ2.

Importantly, we have also added new communities that use ImageJ such computer scientists who were looking for more robust computational solutions.

However the ImageJ ecosystem continues to face major challenges. Perhaps one of the significant is the overall lack of new algorithm development in ImageJ. Relatively few computer scientist and signaling processing folks use ImageJ as a algorithm development or deployment suite. In fact even in our work we many times use Matlab for our algorithm development work. For example in our work with collagen analysis (chapters 3 and 4) and spectral confocal (chapter 7). Even through this matlab code is still open source it is not connected to the major efforts of our group and the broader community in ImageJ2. As discussed in chapter 5 we are taking steps to address such as the development of the Ops framework. This new framework while still too new to see any wide spread adoption. However we also want to encourage people to use the tools that work for them, and not have to rewrite in ImageJ unless advantageous for them.

In scientific analyses we are only as powerful as the tools we use. There is an opportunity cost any time a researcher must choose between one tool and another; if lacking interoperability, a worst-case scenario requires re-implementing a function in one environment when it may be already available in the other. Duplicating efforts like this consumes development time that could be spent elsewhere, creates an ongoing maintenance burden, and may have minimal impact if the result cannot be shared and reused within the community, or if that maintenance cost is not paid. So we are taking great recent efforts to increase interoperability between ImageJ and other analysis suites such as MATLAB. A current major goal is improved interoperability between ImageJ and MATLAB. We are using them as end goals and use cases ImageJ as an open-source, extensible platform built for scientific image analysis, and MATLAB, as a licensed but ubiquitous tool for statistical analysis and algorithm development. We have recently created a ImageJ-MATLAB library that provides a bi-directional bridge for passing data between these two environments, facilitating mutual exchange of algorithms and functionality. Our hope is this bridge will strengthen the bond between the ImageJ and MATLAB communities, to the betterment of the scientific achievement.

Beyond the improvement of these tools in specifically and in general I see two other areas that resonate with this of interoperability.

1) Better Metadata Capture

2) Open Source Hardware

For better metadata capture and sharing the challenge is how best to do this. Ever since the birth of the digital age of imaging, there has been the struggle of how to best represent the binary data and accompanying metadata on disc. This problem has only magnified with a wide array of imaging instruments, assorted methods and plethora of options for local and cloud storage. This has resulted in many scientific imaging formats, the microscopy community use case is especially demonstrative, with over 200 formats in biological microscopy alone. This explosion of formats has resulted in significant issues of file sharing, interoperability and challenges in data retention. As a result in both the medical imaging and biological imaging there have been cries from the user bases to address this. The most common proposed solution to this problem has been the call for a common file format standard for imaging. The argument is to find a binary standard with a accompany data-model that can adequately and accurately contain all the image and pertinent supporting acquisition metadata for a given experiment. The hope is such a file format would be adopted by all the imaging acquisition vendors and by commercial and academic analysis packages. Such a proposal has great merit and has resulted in some large success stories with DICOM in particular as a notable asset in medical imaging where many medical imaging systems directly write this format and many software packages read in the DICOM format. This has resulted in many calls to adopt DICOM further or develop DICOM inspired file format standards for other imaging domains such as biological light microscopy.

After many years of use it has become clear that despite the many strengths of DICOM it is far from a gold standard file format that can represent the medical imaging field with accurate and rigidly held guidelines for how to store and parse image and metadata. Rather a key to DICOM's success has been its ability to adapt, and DICOM has instead become a collection of closely related file standards and formats all with a common base but a number of variants. DICOM also has failed to have significant adoption beyond its original intended user based of medical imaging. This is not a shortcoming of DICOM but rather a direct result of fundamental differences in technology and practices between medical imaging and biological imaging. Instead biological microscopy has attempted to find it own standards with tiff-based variants being a prominent candidate. Many groups have built open and proprietary file formats based on the adobe tiff binary format standard. It also has been a major goal to represent the metadata of acquisition and analysis workflows in open standards that become abridged to the binary file.

In 2002 the LOCI lab at the University of Wisconsin embarked on a simple concept to deal with the burgeon file format challenges faces by the mixture of both home built and commercial systems at the Madison campus and the file format parsing challenges of the lab developed imaging analysis tools under development by LOCI including VisAD and ImageJ. The idea emerged of developing a common library that could be reused by other developers that would parse all available file formats to a common open format. This concept gelled even further in 200X when LOCI became part of the Open Microscopy Environment (OME) consortium. At the time OME was embarking to develop a open source informatics framework for the storage and haring of biological optical microscopy dataset. Notably the goal was not to merely share the image data in a database but instead have a data model based file storage system where the experimental metadata would be parsed and searched. Key to this goal was the development of a data model that would faithfully re-capitulate all the features of a biological acquisition in a open full documented schema. This data model was developed in XML for its ease of parsing and cross-platform sharing. This schema dubbed OME-XML became the heart of the OME database system that evolved into the current OME platform that is now called OMERO. Together OMERO and OME-XML represented for the first time a open freely available system for biological microscopy. Shortly after joining the OME consortium LOCI proposed the addition of this file-format library concept that at this point has evolved into a JAVA based reader of over 50 formats that was called Bio-Formats. As well LOCI proposed that Bio-formats use OME_XML as its metadata models and write all its conversions into a open file format that merged the open TIFF binary format standard with the OME XML metadata standard. The OME consortium formally created this new file format called OME-TIFF that combined the TIFF and XML files into one format by a TIFF format that contained the XML metadata direction in the information header of the TIFF file. The OME-TIFF and Bio-formats quickly became stalwarts of the OME system with all images brought into a installation of OMERO done by Bio-formats based translation. Importantly OMERO also offered direct OME-TIFF based import and export.

Quickly though the Bio-formats approach has become a community wide tool and eclipsed its use as a OMERO import engineer. While the use of Bio-formats in over 1000 OMERO installation has been a important one, Bio-formats quickly began to adapted by a myriad of other software systems both commercial and academic that has similar format challenges. Particularly in the academic arena bio-formats has seen widespread adoption with thousands of labs using and over 20 separate software packages that directly use.

With this this widespread adoption has come serious challenges for Bio-Formats and calls to for new additions. In particular has come the recognition that OME-TIFF is not sufficient in all modality use cases. As well there is significant interest in supporting the writing of more proprietary tiffs. While the consortium has been reluctant to write proprietary standards for fears of propagating proprietary file standards it is also not clear what the best approach.

Our SCIFIO approach as discussed in Chapter 5 partially addresses this problem by generalizing the data model. We think this SCIFIO solution could be adapted to support a myriad of community specific proprietary file format problems and address new problems such as medical image data. However there is a larger issue of what to do about the actual creation of the proprietary file format. For most of our time in developing these readers we have emphasized the support for all file formats in our readers but also stress to developers and users alike the need to eliminate these formats from being used at all. Many groups including ours have advocated the idea of open accessible file formats that are proprietary. However we realize in many cases there are need for new extensions to existing formats and in more rare cases the need for a whole new format to deal with a application type such as a multimodal instrument. So in keeping with our interoperability them where we embrace other applications as long as there are clearly defined APSI for communications, the same concept can be extended to proprietary file formats. As long as the creator of the format is willing to create documented, carefully test file readers and converters many of the issues of the proprietary file format are addressed. Rather than looking for one image format as the community standard we now realize a more practical solution might be multiple open image format standards and tools such as SCIFIO and Bio-Formats to convert between them.

Other challenge and potential opportunity in the theme of interoperability is the concept of open source hardware. As the imaging systems are increasingly becoming compatible with open source solutions whether through the direct acquisition, the file format or image analysis scheme, there is great interest in greater transparency for the entire modality.

Open source software has evolved from being a nice concept in the computer science largely utilized by domain experts to being a widely known concept in many technical areas and even in some consumer and end user arenas with the popularity of major open source projects such as Linux. With the growth of open source in many communities, the sciences have been early and consistent leaders in open source with many scientific software tools being open source. One field that has been major adopters of this freely available sharing concept is biological imaging with the notable example of ImageJ. As discussed in Chapter 2, ImageJ since its advent in almost thirty years ago has emerged as a major tool in the biological sciences and has spawned numerous spin offs and related tools. With these advances in both use and development, it has become desirable to not only make the software that is being used open but also the process by which it is acquired open. There is already a widespread concept for making the protocols for any published study open and carefully documented, but the hardware used for most imaging experiments whether homebuilt or commercial can often be effectively a black box. In this age of the quest for reproducible quantitative microscopy, the open source concept should be applied to the complete imaging system including hardware, not just the software used to analyze the resulting data. We now argue that many of the great advantages of open software can and should be applied to imaging hardware and systems. This is a very new area and will require incentives to develop in this way both in terms of motivating developers to make their technology open source but also demonstrating clear technical advantages in doing so. We plan on devoting more of our resources to this area. One place we see clear technical advantage in open source hardware is in the area of computational optics. Increasingly we are seeing the lines blurred between image analysis and acquisition, where with more regularity we see demand for wanting to run analysis routines at runtime. This can be largely be achieved by calling existing analysis code from scripting frameworks in acquisition engines and doesn't require any in depth knowledge of the hardware. But in computational optics there is the emerging concept that digital approaches can directly influence and manipulate the binary data as it comes and the "raw" collected dataset is actually computationally assisted. There are clear places where a open hardware and open software approach would intersect such as designing a open source all digital microscopy that replace conventional glass optics with digital detection schemes and integrated analysis routines that can help the use achieves goals such as better selectivity for the areas of interest, higher speed etc. We believe that more openness on the methods used in the imaging acquisition and analysis will result not only in ease of sharing and improved reproducibility but also lead to new technical breakthroughs enabled by this openness.