Topic Introduction

Software Tools, Data Structures, and Interfaces for Microscope Imaging

Nico Stuurman and Jason R. Swedlow

The arrival of electronic photodetectors in biological microscopy has led to a revolution in the application of imaging in cell and developmental biology. The extreme photosensitivity of electronic photodetectors has enabled the routine use of multidimensional data acquisition spanning space and time and spectral range in live cell and tissue imaging. These techniques have provided key insights into the molecular and structural dynamics of living biology. However, digital photodetectors offer another advantage—they provide a linear mapping between the photon flux coming from the sample and the electronic sample they produce. Thus, an image presented as a visual representation of the sample is also a quantitative measurement of photon flux. These quantitative measurements are the basis of subsequent processing and analysis to improve signal contrast, to compare changes in the concentration of signal, and to reveal changes in cell structure and dynamics. For this reason, many laboratories and companies have committed their resources to software development, resulting in the availability of a large number of image-processing and analysis packages. In this article, we review the software tools for image data analysis that are now available and give some examples of their use in imaging experiments to reveal new insights into biological mechanisms. In our final section, we highlight some of the new directions for image analysis that are significant unmet challenges and present our own ideas for future directions.

BACKGROUND

A hallmark of scientific experiment is the quantitative comparison of a control condition and a measure of a change or difference after some perturbation. In biology, microscopes are used to visualize the structure and behavior of cells, tissues, and organisms, and to assess changes before, during, or after a perturbation. A microscope collects light from a sample and forms an image, which is a representation of the sample, biased by any contrast mechanisms used to emphasize specific aspects of the sample. For the first 300 years of microscopy, this image was recorded with pencil and paper, and this artistic representation was then shared with others. The addition of photographic cameras on microscopes enabled mass reproduction and substantially reduced, but by no means eliminated, the viewer's bias in recording the image for the first time. Even though the microscope image was directly projected onto the recording medium, the nonlinear response of film to photon flux and its relative insensitivity in low-light applications limited the application of microscopy for quantitative analysis.

DIGITAL IMAGES

What Is a Digital Image?

Microscope digital images are measurements of photon flux across a defined grid or area. They are recorded using either a detector that is an array of photosensitive elements, or pixels, that records a

Adapted from *Live Cell Imaging*, 2nd edition (ed. Goldman et al.). CSHL Press, Cold Spring Harbor, NY, USA, 2010.

© 2012 Cold Spring Harbor Laboratory Press

Cite this article as *Cold Spring Harbor Protoc*; 2012; doi:10.1101/pdb.top067504

whole field simultaneously or a single-point detector that is scanned, usually as a raster, across the sample field to create a full image. The recorded value at each pixel in the image is a digitized measurement of photon flux at a specific point and corresponds to the voltage generated by electrons liberated by photons interacting with the detector surface. Computer software is used to display, manipulate, and store the array of measured photon fluxes as what we recognize as a digital microscope image.

The Multidimensional Five-Dimensional Image

Each array of pixels generates a two-dimensional (2D) image, a representation of the sample. However, it is now common to assemble these 2D micrographs into larger entities. For example, a series of 2D micrographs taken at a defined focus interval can be thought of as a single three-dimensional (3D) image that represents the 3D cell under the objective lens. Alternatively, a series of 2D micrographs taken at a single focal position at defined time intervals form a different kind of 3D image—a time-lapse movie at a defined focal plane. It is also possible to record a focal series over time and to create a four-dimensional (4D) movie. Any of these approaches can be further expanded by recording different contrast methods—the most common, by far, is the use of multiple fluorophores to simultaneously record the concentration of different molecules at the same time. In the limit, this generates a five-dimensional (5D) image. We have chosen to use the singular image to emphasize the integrated nature of this data structure and that the individual time points, focal planes, and spectral measurements all are part of a single measurement.

Regardless of the specific details of an experiment, an image actually has all these dimensions, but some only are of unitary extent. In the simplest case, the recording of a fluorescence signal from a single wavelength and focus position at a specific time generates a 5D image. The focus, time, and wavelength dimension all exist but are just equal to 1. Thus, recording more than one fluorophore simply extends the spectral dimension, as recording a time series extends the time dimension. In this approach, extents change but the intrinsic dimensionality of the image does not. The advantage of this approach is that it provides a single data structure for all data storage, display, and processing. For example, processing of a 5D image only requires definition of focal planes, time points, and wavelengths. In most cases, a single application, aware of the 5D form of the data file, suffices to handle data of different extents.

One of the most difficult parts of working with these data structures is the lack of a defined nomenclature for referring to the data. Images that sample space only are sometimes referred to as “3D images” or “stacks.” Time-lapse images are often referred to as “movies” or “4D images.” Time-lapse data can be stored in their original format or compressed and assembled into a single file and stored in proprietary formats—QuickTime, AVI, WAV, etc. These compressed formats are convenient in that they substantially reduce the size of files and are supported by common presentation software (e.g., PowerPoint and Keynote), but they do not necessarily retain the pixel data in a form that preserves the integrity of the original data measurements. It is important to be aware of the distinction between compression methods that are lossless (i.e., the original data can be restored) and lossy (often much better at reducing storage but losing the ability to restore the original data).

Monochrome versus Color

Microscope images can be either monochrome or color, and it is critical to know the derivation and description of the data in an image to understand what is actually being measured. Monochrome images are single-channeled images and are the most direct mapping of the photon flux measurements required by the photoelectronic detector. They are used as the basis of more elaborate displays using color to encode different channels or different lookup tables. Color images may be created based on the display of multiple monochrome images; however, images may also be stored as color (e.g., JPEG). Analysis of color images (e.g., images of histology sections) is possible but often starts by decomposing the color image into the individual RGB (red–green–blue) channels and by processing them separately. Analysis based on differences in intensity should be undertaken with caution, as the files that store color images rarely retain the quantitative mapping of photon flux measured by the detector.

Bit Depth

When stored in digital form, data from electronic detectors are stored in bits, or more formally, a base-2 representation of the quantitative measurements from the photodetector. By convention, a byte is a sequence of 8 bits and thus can represent numerical values from 0 to $2^8 - 1$ or 255. Data that can be represented in this range are referred to as 8-bit data and have a bit depth of 8 bits. Most scientific-grade charge-coupled device (CCD) cameras digitize their data to either 12 bits (data range from 0 to $2^{12} - 1$ or 4095) or 16 bits (data range from 0 to $2^{16} - 1$ or 65,535). When stored in computer memory, 8-bit data map easily to a single byte, whereas 12-bit or 16-bit data must be stored in 2 bytes in sequence to properly represent the data. In general, data-acquisition software handles storage of these data without any intervention from the user. However, when moving data between different software programs, unexpected transformations can occur. For example, data recorded with a 12-bit CCD camera will appear as 16-bit data to any visualization or analysis program that reads it. Most microscopy software tools handle this difference properly; however, some (such as Photoshop) display an image assuming a possible dynamic range of 2^{16} , requiring the user to manually change the display settings. For these reasons, knowing the bit depth of one's data is helpful in understanding what is actually displayed.

Metadata (Nonpixel Data)

A critical component of any microscope digital image is the image “metadata”—that is, the nonpixel data that describe the image pixel data or binary data. Metadata can include a large number of measurements and quantities about the image. The most important are the dimensions of the image, the bit depth or number of bits per pixel, and the physical size of each pixel in the sample; there are a large number of others. These are the most basic forms of metadata and largely relate to the image itself. Proper display and analysis of the 5D image described above depends on a known specification that records the extent of each of the dimensions and as much information about sampling intervals, descriptions of spectral properties, imaging contract modes, etc., as possible. In addition, many imaging files also store information describing the image acquisition, detailing the settings on the microscope used to record the image. In general, image metadata are a critical component of the image and are certainly mandatory for systematic use of software tools for image display and analysis.

Proprietary Formats

Microscope digital images must be written as a file on a file system, and this file is written in a defined specification or file format so that the data can be read by another software program. File formats include facilities to store the image data themselves as well as a selection of image metadata defining aspects of the image and the data-acquisition process. Almost all packages support the concept of 5D data, as described above, but the details of file types used are quite variable: Some store the data in a single file, whereas some use a directory on a file system to store all the individual frames of an image in separate files. Many of these derive from commercial software used to run commercial turnkey acquisition systems. Each commercial software package uses its own proprietary file format, and the rapid growth in imaging products has spawned a large number of image file formats. Our count has identified at least 60 different formats used in biological microscopy. This creates a significant problem for anyone wanting to transfer data written between software packages or operating systems.

File Format Tools: Bio-Formats

Knowing the correct image, metadata are essential for working with images for visualization and processing. The metadata describe fundamental details such as the size of the image, the position of the origin, and the number of bytes in the image file in each pixel and also can contain critical information

about experimental settings (e.g., the optical section thickness); thus, having access to the metadata associated with any image is critical for properly analyzing and viewing microscope images.

Most software write image data and metadata in their own format, and this has led to a plethora of image file formats across life sciences microscopy. Many of these derive from commercial software for which each commercial package uses its own preparatory file format; in addition, there are a small number of public projects developing image-processing software for biological microscopy, and, again, many of these have their own file format. This creates a significant problem for anyone wanting to access data written by one software package in another software package; this lack of standardization continues to plague the field. By far, the best way to deal with this problem is to use a publicly available library that can translate a large number of preparatory file formats into a standardized data structure available to essentially any software. This library, called Bio-Formats (<http://www.openmicroscopy.org/site/products/bio-formats>), is the result of an open project founded at the University of Wisconsin in Madison and by Glencoe Software Incorporated. The library is written in Java and is an open-source and open-development resource publicly available under the GPL (General Public License) (<http://www.gnu.org/copyleft/gpl.html>). As of this writing (mid-2009), this library reads image metadata and data for 70 different open and proprietary file formats and can be used within ImageJ, MATLAB, and many other popular image-processing programs.

Standardized Formats

The panoply of microscopy image file formats creates problems for the scientist who requires data written with one software package to be read by another software package. Moreover, as imaging becomes more of a mainstay of systems biology, the development and applications of new processing algorithms is hampered by the burden of supporting many different file formats. For these reasons, a standardized format that can be used by all software tools is required. This format must capture the image metadata using a commonly accepted specification and also store the binary data (the actual pixels) in a commonly accessible form.

In 2005, the Open Microscopy Environment (OME; <http://openmicroscopy.org>) proposed an XML (Extensible Markup Language)-based metadata specification known as the OME Data Model (Goldberg et al. 2005) (<http://ome.xml.org>). This specification provides a mechanism for describing most common modes of microscopy (fluorescence, phase, differential interference contrast) and can record imaging system parameters for most imaging methods (wide field, confocal, multiphoton). The metadata and binary image data can be written as a stand-alone format (OME-XML) that captures a full 5D image in a single XML document with the binary image data stored as compressed base 64. This is convenient for transport but performs poorly for analysis and visualization. An alternative is to use the metadata specification in OME-XML and to store this in the header of a TIFF (Tagged Image File Format) file. This hybrid file format is known as OME-TIFF and has a number of distinct advantages.

- Image planes are stored within one multipage TIFF file or across multiple TIFF files. Any image organization is feasible.
- A complete OME-XML metadata block describing the image is embedded in each TIFF file's header. Thus, even if some of the TIFF files in a 5D image are misplaced, the metadata remain intact.
- The OME-XML metadata block in an OME-TIFF file is identical to the metadata contained in an OME-XML file and can refer to pixel data stored in single or multiple TIFF files.
- The only conceptual difference between OME-XML and OME-TIFF is that instead of encoding pixels as base-64 chunks within the XML such as OME-XML does, OME-TIFF uses the standard TIFF mechanism for storing one or more image planes in each of the constituent file(s).
- Storing image data in TIFF is a de facto standard, and essentially all image-handling software can read TIFF-based formats; thus, adoption and integration of OME-TIFF is straightforward.

A full description of the OME Data Model (Goldberg et al. 2005) may be obtained at the website <http://ome.xml.org>. OME-XML and OME-TIFF are supported by a number of commercial software providers and can be used as an export format from ImageJ and MATLAB (see below).

DATA ACQUISITION

Acquiring a Digital Image

Digital microscope images can be acquired either by projecting the magnified image of the object directly onto the camera or by scanning a sample and measuring light intensity at each spatial element using a fast detector such as a photomultiplier tube or avalanche photodiode. In both cases, intensity data end up in computer memory, organized in rows and columns, from where they can be displayed, stored, or otherwise manipulated. When using scanning for image formation, control of the scanning device (which often consists of mirrors mounted on a galvo) needs to be coordinated with timing of the intensity measurements such that the computer has knowledge of which intensity measurement belongs to which spatial element.

Maximizing Information Content

The information content of the acquired images is determined by the intensity of the captured photon flux (signal), noise in the signal, and noise in the detection system. Noise in the signal itself is called shot noise and increases with increasing signal intensity. However, because shot noise increases more slowly than signal strength itself, increasing the signal strength will increase the signal-to-noise ratio (SNR). Noise in the detection system most often consists of a signal-independent constant (lowering this constant is worth a premium!) and possibly a component linearly dependent on signal strength. Therefore, detector-system-generated noise can also be overcome by increasing the signal strength. These considerations make it desirable to acquire images in which the maximum detected signals are close to the saturation point of the detector. It is important that the saturation of the detector not be surpassed, as such clipping will result in an unreliable representation of the object. So, how does one go about increasing the detected signal?

Obvious ways to do so are by increasing the photon flux itself. This can usually be accomplished by increasing intensity of the illumination source. One can also increase the integration time on the detector (by increasing exposure time of the camera or by scanning more slowly in a raster-scanning system). A drawback of both methods is the increased photon dose on the sample increasing the likelihood of photon-induced damage and photobleaching. Moreover, increasing integration time might lead to motion blur in live samples. Thus, information content of live cell images is almost always limited by sample-imposed constraints on photon dose. Another approach is to sacrifice spatial resolution, for instance, by binning multiple pixels on a camera. In all cases, it is essential to optimize the light path between the sample and the detector such that minimal light loss is incurred. This includes the use of objective lenses with high numerical aperture in fluorescence imaging so that a larger fraction of the light emitted by the sample is collected, reducing optical aberrations such as spherical aberration as well as optimizing the design of optical filters and/or other elements to maximize light throughput.

Autofocus

When images are taken at multiple positions, and/or when there is focus drift during the time frame of acquisition (caused, for instance, by slight changes in environmental temperature), an autofocus mechanism is needed. There are two principal methodologies for autofocus used in microscopy: image-based autofocus and reflection-based autofocus. In image-based autofocus, a series of pictures is taken at different focus positions, and the position of best focus is calculated based on image content (Rabut and Ellenberg 2004). The methodology consists of assigning a focus factor to each acquired image, for instance, by finding maximum intensity, sharp edges, or using other image-analysis

techniques and then estimating the position of best focus from the relation of focus position and focus factor. In reflection-based autofocus systems, a (often infrared) light beam is directed through the objective toward the image. This light beam is reflected back (in the case of air objectives, mainly from the air–coverslip interface; for oil-immersion objectives, from the coverslip–cell medium interface) and detected by a sensor. By combining measurement of this reflection signal with movement of the focus stage, it is possible to find such a reflecting surface and/or to lock onto that surface (employing a direct feedback loop between sensor and focus stage). Currently available implementations differ in their ability to apply an offset between the point of optical focus and the reflecting surface as well as the speed of the feedback loop between sensor and focus drive. Image-based autofocus is often much slower than reflection-based autofocus and also results in exposure of the sample to illuminating light (rather than the more benign and lower-dose infrared light used in reflection-based autofocus), which can result in phototoxicity and photobleaching. However, when the distance between sample and surface varies, reflection-based autofocus by itself cannot be satisfactory. Development of robust autofocus mechanisms has driven automated high-content/high-throughput image-based screening technologies, and the increasing availability of autofocus on research microscopes is fueling automated high-throughput approaches in live cell imaging.

Computer Control of Imaging Devices and Peripherals

Current research microscopes are equipped with a multitude of computer-controllable components such as x - y stages, focus drives, shutters, filter wheels, etc. These motorized components make possible the fully automated acquisition of 5D images. First, the user will specify a protocol that defines the desired sequence of component movements and image acquisition. For instance, to obtain a 5D image using a camera-equipped wide-field epifluorescence microscope, software will move the x - y stage to the desired position, move the z drive to the start position, put the desired dichroic mirror, excitation and emission filters in place, open the correct shutter, start exposure of the camera, close the shutter, read out the image from the camera, move another dichroic mirror and filters in place, open the shutter, start camera exposure, close the shutter, read out the image, move the z drive to the next position, and so forth and so on. It is the task of acquisition software to carry out such sequences of instrument-control events as quickly and as reproducibly as possible yet be flexible and easy to configure for the end user.

Software Tools for Image Acquisition

Many microscope image-acquisition software packages are available. In those cases where the interface to the equipment is proprietary, acquisition software can only be provided by the hardware vendor. This is currently almost universally the case with commercial scanning (confocal) microscopes (a notable exception is the software ScanImage from the group of Karel Svoboda at Janelia Farm). Camera-based systems are often more open, and interface descriptions are available for many motorized microscopes and scientific-grade cameras, as well as for most peripheral equipment such as shutters, filter wheels, and stages. Accordingly, a number of third-party software packages for camera-based microscope image acquisition are available (examples are MetaMorph from Molecular Devices, Image-Pro Plus from Media Cybernetics, Volocity from Improvision/Perkin-Elmer, and Slide-Book from Intelligent Imaging Innovations, Inc.). In addition, there is a growing trend among microscope companies to produce software that only works with their own microscopes (examples are AxioVision from Carl Zeiss and NIS-Elements from Nikon).

In all cases, the number of different types of microscope equipment supported by each software package is limited because of the large variety of such equipment and the lack of computer interface standards. Also—with very few exceptions—it is impossible for third parties to add support for a microscopy-related device to an existing software package. Most of the software packages only work within the Microsoft Windows operating system, and programmatic access to them is usually highly limited, causing advanced imaging laboratories to not use these packages but to code their own in programming environments such as LabVIEW. Because of these constraints, it is typical in

research–laboratory environments to have different software packages run each microscope system, drastically increasing training requirements and frustrating researchers who spend more time becoming familiar with a particular user interface rather than the principles of operating a microscope. Because of the low volume of sales in comparison to consumer software, the price of image-acquisition software is relatively high (\$5000–\$15,000), and quality can be disappointing.

To alleviate these issues, a group (including one of the authors) in the laboratory of Ron Vale at the University of California, San Francisco started development of open-source software for microscope control named μ Manager (<http://micro-manager.org>) in 2005. The software is cross-platform (runs on Windows, Mac OS X, and Linux), has a simple user interface, and allows for programmatic extensions in many different ways. It now supports a large number of devices used in camera-based microscopy imaging. Most importantly, μ Manager has an open programming interface to devices, allowing anyone to add support for a novel device to μ Manager. A significant number of such “device adapters” have already been contributed by third parties. This programming interface also allows for programmatic abstraction of devices such as cameras, shutters, and stages so that programs or scripts written for the μ Manager Core can now work with all hardware supported by μ Manager, greatly expanding their usefulness. Not only does this give advanced-imaging laboratories a way to easily make their developments quickly available to other scientists, but also it gives microscopists a common interface for microscope equipment that will hopefully grow into an industry-wide standard, reducing cost and increasing quality of software for microscopy.

IMAGE PROCESSING

Image-Processing Fundamentals

Image processing is a broad term that encompasses a large number of different kinds of image manipulation. In general, image-processing routines can be characterized as either linear or nonlinear. Linear approaches retain the proportionality of relative intensities, whereas nonlinear approaches potentially may change the relative intensities. For this reason, intensity-based measurement and analysis can only be performed after processing with linear-based approaches. However, nonlinear approaches are often quite useful for segmenting images to create masks that can be used to define the boundaries of images for signal quantification.

Nonlinear Contrast Enhancement

There are a range of nonlinear approaches for enhancing contrast in images. The best comprehensive resource for the schemes is John Russ’s thorough presentation of image processing (Russ 2007). These range from edge detection schemes to convolution techniques and other filtering techniques. It is important to distinguish between approaches that only work on 2D images and those that consider the full 3D volume. In general, these approaches are used either to generate a representation of data that is easier to appreciate visually or to generate masks to use for further quantification of the original image.

Deconvolution

Deconvolution techniques make use of knowledge of the point-spread function of the microscope to improve the SNR and contrast in an image. There is a wide variety of deconvolution techniques that have been implemented in most open and commercial image-processing packages. In general, deconvolution approaches can be classified as either deblurring techniques or restoration techniques. Deblurring techniques use the point-spread function to estimate blurring and subtract the blurring from the original image. By contrast, restoration techniques also use the point-spread function but attempt to calculate the distribution of intensity in the sample based on the point-spread function. Restoration techniques are usually iterative calculations and therefore require substantial processing

power. Full descriptions of these techniques and their applications in biological microscopy are available (Swedlow et al. 1997; Wallace et al. 2001).

Image-Processing Platforms

A wide variety of commercial and open image-processing platforms are available for processing biological images. Photoshop is a standard image-processing tool that is often used for simple image enhancement, cropping, and color-mapping changes. Photoshop only handles 2D images and is a very powerful application; unfortunately, Photoshop can also be used to substantially change the appearance of an image. It is important to remember that the images that are being processed are actually data, and the original appearance of the data must be preserved. A very clear and definitive definition of appropriate uses of imaging processing in cell and developmental biology has been published by *The Journal of Cell Biology* (Rossner and Yamada 2004).

There are a number of commercial image-processing packages that are dedicated toward biological microscopy. These are too numerous to delineate here but are available from most vendors' websites. In almost all cases, these handle multidimensional time-lapse and 3D images or the 5D image and also provide sophisticated image-processing and visualization tools. To complicate matters, all commercial image-acquisition software comes bundled with substantial image-processing capabilities. These commercial packages have varying support for file formats beyond their own. The user should compare the ability to move data within these packages carefully.

A very commonly used open platform for image processing in microscopy is ImageJ. ImageJ is an open Java-based program that has pluggable architecture and, for this reason, has become a popular tool used for most noncommercial image processing and analysis. ImageJ is free to download (<http://rsbweb.nih.gov/ij/>) and includes an extensive library of plug-in functions that are developed by the community and extend the basic functionality of ImageJ (for instance, the aforementioned acquisition software μ Manager can run as an ImageJ plug-in). ImageJ can be used for image acquisition, image analysis, image processing, and final production of figures. ImageJ has been maintained by Wayne Rasband (National Institutes of Health [NIH]). There are now a number of efforts going forward to extend the architecture and functionality of ImageJ using modern software programming techniques.

ANALYSIS

Object Definition and Measurement

Analysis of biological image data usually proceeds by identifying the objects to be measured and then actually measuring their properties. Identification of objects is called segmentation. There is a wide range of segmentation tools available, and most image-processing packages have many different choices for methodologies to define the boundaries of objects. As noted above, it is important to know whether the algorithm used is working in 2D or 3D. A full description of segmentation algorithms is available. In general, segmentation methods use some method of defining an object based on intensity boundaries but can include more sophisticated measurements. Almost always, the parameters used for segmentation are empirical and defined by a user's knowledge of the objects they want to measure. The end product of segmentation is a defined object in space and/or time with defined boundaries of what is included in the objects and what is excluded. Thus, segmentation is the basis for further analysis of object intensities and object tracking.

For single-molecule imaging, a standard approach is the fitting of a Gaussian-shaped function to every bona fide molecule measured in the image. This fit image is then used for further characterization.

For live cell imaging, a very common analysis is the tracking of objects as they move through space across time. A wide variety of tracking algorithms is available. The simplest ones take an object at time t and identify its nearest neighbor at time $t + 1$. These neighbor-based techniques (e.g., Platani et al. 2002) are appropriate only for situations in which there is a small amount of movement between

frames and there are fairly sparsely distributed objects in the image. More sophisticated tracking tools are necessary for more difficult images. One tool that one of us (J.R.S.) has used successfully combines a global-minimization approach and quite accurate gap filling to produce powerful tracking performance (Jaqaman et al. 2008).

Object and Measurement

Once objects are defined, a wide range of measurements can be performed across them. Most commonly, measurement of the intensity contained within an object is calculated and then corrected by the background intensity found around the object. More sophisticated analysis includes measurement of the lifetime of the fluorophore within the object. Fluorescence lifetime measurements are quite sensitive to changes in the environment of a fluorophore and are increasingly used for analysis of Förster resonance energy transfer (Wouters et al. 2001).

A large number of parameters can also be calculated on objects that characterize the shape of the object and the distribution of intensity within the object. Shaped parameters include the measurement of elongation, skew, kurtosis, and deviations from an ideal circle. The distribution of intensities within an object is often referred to as textures and can be calculated by fitting classes of polynomials to the object. There are a large number of different types of features that can be calculated for each object, and often, defining the calculations to be performed is an empirical task to identify what derived parameter can be used in an image-based measurement.

Machine Learning Methods

Over the past 10 years, a number of groups have applied well-established machine learning tools to the distribution of signals within microscope images. The first studies by the Murphy laboratory demonstrated the potential application of this approach by identifying subcellular distributions that humans could not even visualize (Boland and Murphy 2001). Since then, a number of groups have helped develop these concepts and have applied them to a large number of biological problems from single-cell analysis to histological sections to high-content screening (Neumann et al. 2006; Zhou and Peng 2007; Orlov et al. 2008). All of these methods calculate a large number of intensity textures and shaped-based features and then use various well-established methods for defining which features can discriminate between multiple different classes. These tools can potentially be quite powerful and can be used to define classes that might not be obvious to the user.

Tools for Image Analysis

There are a large number of commercial and open packages for multidimensional image analysis. ImageJ is an open framework that includes plug-ins for quite powerful image analysis. In addition, the MATLAB and IDL (Interface Definition Language) frameworks provide scripting frameworks for sophisticated image analysis. All of these tools are commonly used in modern biological research. Again, these very powerful tools should always be used with caution to ensure that no perturbations are caused in the data.

Critical Applications for Image Analysis

Image processing and analysis are now ubiquitous in cell and developmental biology. We have described the basic tools that are available in most commercial and open image-processing packages and used for most analysis. However, there are a number of much more sophisticated tools that deserve mention as examples of what is possible when a careful image acquisition and advanced mathematics are brought to bear.

The first example is fluorescent-speckle microscopy. This approach is used to characterize the dynamics of polymers, especially the cytoskeleton systems in living cells (Danuser and Waterman-Storer 2006). The approach has been especially powerful for analyzing the dynamics of microtubules and actin in the mitotic spindle and during cell motility. These methods depend on the use of

advanced image acquisition and object identification and tracking. The Danuser and Waterman laboratories have pioneered the application of these techniques and have revealed substantial new understanding of the properties and dynamics of the cytoskeleton.

One of the most powerful uses of live imaging is the measurement of the dynamics of molecules or macromolecular complexes in living cells. With the maturation of laser-scanning confocal microscopes, it became a routine to monitor the movement of fluorescently labeled molecules using fluorescence recovery after photobleaching. This method was first applied to measure the dynamics of molecules into 2D monolayers (Axelrod et al. 1976), but it has been extended to analyze molecular dynamics in a wide variety of cells and tissues (Lippincott-Schwartz et al. 2001). The method measures the recovery of fluorescent protein into a bleached region. The most common reported values are the $t_{1/2}$ of the recovery and the mobile fraction or the fraction of the labeled protein that is mobile on the timescale being measured. More sophisticated analyses can reveal the characteristic of binding of the fluorescent protein to its receptors and the presence of multiple forms of the protein with distinct mobilities (Braga et al. 2004; Rabut et al. 2004; Sprague and McNally 2005). These approaches provide a quantitative basis to compare molecular association in living cells.

The use of imaging for large-scale assays is becoming more and more routine. A large number of small interfering RNA screens have revealed a substantial number of new components of critical cellular machinery and new pathways and networks. Signaling and motility have been examined in *Drosophila* S2 cells (Ramadan et al. 2007), and the machinery that is responsible for driving the function of the mitotic spindle has been extensively examined using polygenome knockdown in *Drosophila* and in human cells (Bettencourt-Dias et al. 2004; Neumann et al. 2006; Goshima et al. 2007). These studies have revealed a substantial number of new components of the mitotic spindle. In all cases, new combinations of object identification, feature selection, and machine learning have been used to identify new components and pathways. These assays stand as a testament of the kind of sophisticated analysis that is possible when advanced image processing is brought to bear in biological imaging.

IMAGE DATA MANAGEMENT

In our own laboratories, it is quite common for a single graduate student or postdoctoral fellow to generate >500 GB of images during their time in our laboratories, and, inevitably, they find themselves struggling to manage the collection of image data files associated with their experiments. For this reason, image data management has become a critical issue that can hinder biological discovery. Data can be organized on the file systems that are in common use in all laboratories or by using specific data-management applications. In this section, we review these facilities and their strengths and weaknesses.

Data Management on the File System

Storing data on the file system is perhaps the simplest way to manage large volumes of image data. It allows users to make use of very familiar tools such as file names and directories to organize and track data. For sophisticated multistep analysis workflows, simply adding specified monikers to file names or putting results files in specific directories is often sufficient to identify data files. However, file-system-based data management does not usually allow multiple users to visualize and, for example, annotate and analyze data on the file system, simply because the tools that are needed to track who does what to each file are not available. Most important, access to the data depends on directly accessing the file system, which may not be possible for users or collaborators who do not have sufficient privileges or who are not within a laboratory's firewall.

Data-Management Applications

An alternative to using the file system involves building applications for the management of large-scale data sets from imaging experiments. In general, these applications are based on a server–client architecture in which an application is built and runs on a central server and then delivers information out

to client applications connecting through the Ethernet. Most commonly, those client applications involve web browsers, but certainly, image clients can be built in any programming language. The use of this type of architecture enables a single-server application to serve multiple users and to provide remote access in a flexible and easily accessible way.

Server/client applications are usually built as a series of layers, often in a so-called three-tier architecture. The foundation or bottom of the application is a database application that stores data and the links between different data elements using a relational database. At the next level, a middleware application provides the definitions and the policies to communicate with external clients, thus delivering the relational database to the so-called front-end or client applications. These provide a user-facing tool that enables the use of the data held in the database. The server–client relationships are at the heart of most data-management applications, and the overall design and tools to build them are well standardized and are available quite freely as open-source applications.

Both authors have been involved in software projects that deliver data-management tools for imaging in microscopy. The Scientific Image DataBase (SIDB; <http://sidb.sourceforge.net/>) was a first-generation open-source data-management tool built with help from Nico Stuurman that uses PHP scripts communicating with a PostgreSQL database and a web-browser-based front-end user. This first-generation application showed the usefulness of data management but also demonstrated the difficulty of developing a full-fledged data-management program for images that supported large numbers of file formats and provided facilities for both viewing and managing and analyzing large repositories of image data.

In 2001, Jason R. Swedlow, along with cofounders Ilya Goldberg and Peter Sorger, initiated the founding of the OME (<http://openmicroscopy.org>). OME is an open-grant-funded software development project to build data-management tools for life sciences imaging. OME has released the Bio-Formats library described above and has released a series of applications for image data management in the biological sciences. The basic principles of OME are to develop tools that allow interoperability between software applications. Rather than build specific applications for any scientific domain, OME focuses its efforts on applications that are as generic as possible and serve as interfaces between existing software tools or provide interfaces for future tools to use. In addition, OME releases specifications for common file formats to promote data sharing between individuals and individual software applications.

Since 2000, OME has released data-management applications to support working with large numbers of image and large-image sets as well as supporting interfaces for analysis (Swedlow et al. 2009). From 2000 to 2005, OME released the OME Server, a Perl application for image data management. Since 2007, OME has released Open Microscopy Environment Remote Objects (OMERO), a Java Enterprise application for image-data management. All OME data-management applications rely on a server-client architecture such that a database and middleware application provide all of the storage, referencing and image data, and metadata access and allow connections to the server from client applications running on the user's desktop or laptop computers using a standard Internet connection. OMERO is built to provide a rapid extensible and scalable data-management solution that supports access from a wide variety of programming and application environments. Based on Java, OMERO runs on all major operating systems, is now running in hundreds of laboratories worldwide, and is available for download at <http://openmicroscopy.org>.

CONCLUSION

Modern microscopy, and live cell imaging in particular, requires the use of software tools for data acquisition, processing, and analysis. In many cases, a result from an imaging experiment is only apparent after a significant number of steps of processing and analysis. The technology available to the modern microscopist is developing rapidly, and the arrival of open software tools will provide much more flexible environments for scientific investigation and discovery. There is an ongoing evolution across the whole domain of commercial and open imaging software tools, allowing for

increasingly sophisticated experiments and deeper insights into cellular mechanisms and pathways. We look forward to these advances and the discoveries they will enable.

ACKNOWLEDGMENTS

Work by N.S. on μ Manager is supported by grant 1R01RB007187 from the NIH. Work in the J.R.S. Laboratory is supported by the Wellcome Trust (085982), Cancer Research UK (C303/ A5434), and the Biotechnology and Biological Sciences Research Council (BB/G01518X/1).

REFERENCES

- Axelrod D, Koppel DE, Schlessinger J, Elson E, Webb WW. 1976. Mobility measurement by analysis of fluorescence photobleaching recovery kinetics. *Biophys J* 16: 1055–1069.
- Bettencourt-Dias M, Giet R, Sinka R, Mazumdar A, Loc WG, Balloux F, Zafropoulos PJ, Yamaguchi S, Winter S, Carthew RW, et al. 2004. Genome-wide survey of protein kinases required for cell cycle progression. *Nature* 432: 980–987.
- Boland MV, Murphy RF. 2001. A neural network classifier capable of recognizing the patterns of all major subcellular structures in fluorescence microscope images of HeLa cells. *Bioinformatics* 17: 1213–1223.
- Braga J, Desterro JM, Carmo-Fonseca M. 2004. Intracellular macromolecular mobility measured by fluorescence recovery after photobleaching with confocal laser scanning microscopes. *Mol Biol Cell* 15: 4749–4760.
- Danuser G, Waterman-Storer CM. 2006. Quantitative fluorescent speckle microscopy of cytoskeleton dynamics. *Annu Rev Biophys Biomol Struct* 35: 361–387.
- Goldberg IG, Allan C, Burel J-M, Creager D, Falconi A, Hochheiser HS, Johnston J, Mellen J, Sorger PK, Swedlow JR. 2005. The Open Microscopy Environment (OME) Data Model and XML file: Open tools for informatics and quantitative analysis in biological imaging. *Genome Biol* 6: R47.
- Goshima G, Wollman R, Goodwin SS, Zhang N, Scholey JM, Vale RD, Stuurman N. 2007. Genes required for mitotic spindle assembly in *Drosophila* S2 cells. *Science* 316: 417–421.
- Jaqaman K, Loerke D, Mettlen M, Kuwata H, Grinstein S, Schmid SL, Danuser G. 2008. Robust single-particle tracking in live-cell time-lapse sequences. *Nat Methods* 5: 695–702.
- Lippincott-Schwartz J, Snapp E, Kenworthy A. 2001. Studying protein dynamics in living cells. *Nat Rev Mol Cell Biol* 2: 444–456.
- Neumann B, Held M, Liebel U, Erfle H, Rogers P, Pepperkok R, Ellenberg J. 2006. High-throughput RNAi screening by time-lapse imaging of live human cells. *Nat Methods* 3: 385–390.
- Orlov N, Shamir L, Macura T, Johnston J, Eckley DM, Goldberg IG. 2008. WND-CHARM: Multi-purpose image classification using compound image transforms. *Pattern Recogn Lett* 29: 1684–1693.
- Platani M, Goldberg I, Lamond AI, Swedlow JR. 2002. Cajal body dynamics and association with chromatin are ATP-dependent. *Nat Cell Biol* 4: 502–508.
- Rabut G, Ellenberg J. 2004. Automatic real-time three-dimensional cell tracking by fluorescence microscopy. *J Microsc* 216: 131–137.
- Rabut G, Doye V, Ellenberg J. 2004. Mapping the dynamic organization of the nuclear pore complex inside single living cells. *Nat Cell Biol* 6: 1114–1121.
- Ramadan N, Flockhart I, Booker M, Perrimon N, Mathey-Prevot B. 2007. Design and implementation of high-throughput RNAi screens in cultured *Drosophila* cells. *Nat Protoc* 2: 2245–2264.
- Rossner M, Yamada KM. 2004. What's in a picture? The temptation of image manipulation. *J Cell Biol* 166: 11–15.
- Russ JC. 2007. *The image processing handbook* CRC, Boca Raton, FL.
- Sprague BL, McNally JG. 2005. FRAP analysis of binding: Proper and fitting. *Trends Cell Biol* 15: 84–91.
- Swedlow JR, Sedat JW, Agard DA. 1997. Deconvolution in optical microscopy. In *Deconvolution of images and spectra* (ed. Jansson PA), 284–309. Academic Press, New York.
- Swedlow JR, Goldberg IG, Eliceiri KW. 2009. Bioimage informatics for experimental biology. *Annu Rev Biophys* 38: 327–346.
- Wallace W, Schaefer LH, Swedlow JR. 2001. A workingperson's guide to deconvolution in light microscopy. *BioTechniques* 31: 1076–1097.
- Wouters FS, Verveer PJ, Bastiaens PI. 2001. Imaging biochemistry inside cells. *Trends Cell Biol* 11: 203–211.
- Zhou J, Peng H. 2007. Automatic recognition and annotation of gene expression patterns of fly embryos. *Bioinformatics* 23: 589–596.

Cold Spring Harbor Protocols

Software Tools, Data Structures, and Interfaces for Microscope Imaging

Nico Stuurman and Jason R. Swedlow

Cold Spring Harb Protoc; doi: 10.1101/pdb.top067504

Email Alerting Service

Receive free email alerts when new articles cite this article - [click here](#).

Subject Categories

Browse articles on similar topics from *Cold Spring Harbor Protocols*.

[Image Analysis](#) (124 articles)

[Imaging/Microscopy, general](#) (579 articles)

To subscribe to *Cold Spring Harbor Protocols* go to:
<http://cshprotocols.cshlp.org/subscriptions>