



HAL
open science

Bioimage Analysis Tools

Kota Miura, Sébastien Tosi, Christoph Möhl, Chong Zhang, Perrine Paul-Gilloteaux, Ulrike Schulze, Simon F Norrelykke, Christian Tischer,
Thomas Pengo

► **To cite this version:**

Kota Miura, Sébastien Tosi, Christoph Möhl, Chong Zhang, Perrine Paul-Gilloteaux, et al.. Bioimage Analysis Tools. Kota Miura. Bioimage Data Analysis, Wiley-VCH, 2016, 978-3-527-80092-6. hal-02910986

HAL Id: hal-02910986

<https://hal.science/hal-02910986v1>

Submitted on 3 Aug 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

2

Bioimage Analysis Tools

Kota Miura,¹ Sébastien Tosi,² Christoph Möhl,³ Chong Zhang,⁴ Perrine Paul-Gilloteaux,^{5,6} Ulrike Schulze,⁷ Simon F. Nørrelykke,⁸ Christian Tischer,⁹ and Thomas Pengo¹⁰

¹European Molecular Biology Laboratory, Meyerhofstraße 1, 69117 Heidelberg, Germany
National Institute of Basic Biology, Okazaki, 444-8585, Japan

²Institute for Research in Biomedicine (IRB Barcelona), Advanced Digital Microscopy, Parc Científic de Barcelona, c/Baldiri Reixac 10, 08028 Barcelona, Spain

³German Center of Neurodegenerative Diseases (DZNE), Image and Data Analysis Facility (IDAF), Core Facilities, Holbeinstraße 13–15, 53175 Bonn, Germany

⁴Universitat Pompeu Fabra, Carrer Tànger 122–140, Barcelona 08018, Spain

⁵Institut Curie, Centre de Recherche, Paris 75248, France

⁶Cell and Tissue Imaging Facility, PICT-IBiSA, CNRS, UMR 144, Paris 75248, France

⁷The Francis Crick Institute, Mill Hill Laboratory, The Ridgeway, Mill Hill, London NW7 1AA, UK

⁸ETH Zurich, Scientific Center for Optical and Electron Microscopy (ScopeM), Image and Data Analysis (IDA) Unit, HPI F15, Wolfgang-Pauli-Strasse 14, 8093 Zurich, Switzerland

⁹EMBL Heidelberg, Advanced Light Microscopy Facility, Meyerhofstraße 1, 69117 Heidelberg, Germany

¹⁰University of Minnesota, University of Minnesota Informatics Institute, Cancer and Cardiovascular Research Building, 2231 6th St SE, Minneapolis, MN 55445, USA

2.1

Overview of Bioimage Analysis Tools

In this chapter, we provide an overview of tools useful for bioimage analysis. The list is far from being exhaustive, but most of these are tools we use, or have used, in actual research projects. A few of the listed tools have been tested, but not used, by the authors; they are listed here for their potential and specific features that are not commonly found in other tools.

We grouped the tools into four different categories. Two major categories are the *graphical user interface* (GUI) and the *command line interface* (CLI). The computation engines of these tools are usually very similar (algorithms, libraries), but GUIs allow you to operate on images using menu selection, wizards, and mouse clicks. They are usually easier to get a grip on and also, as images are often displayed during processing, parameter tweaking and algorithm updates can be immediately confirmed by eye, in an interactive way. This allows the user

to manually select regions of interest and easily try out various algorithms to visually track their effects.

In CLI software, the user inputs text commands to process and analyze images. This interface needs some knowledge on the Unix-style command line language, file system, and names of image processing algorithms, to send commands via the terminal.

Since images are not necessarily displayed, the CLI-based image processing and analysis is less intuitive than that based on GUI and it adds a hurdle for the beginner. However, the advantage is that once the user has learned this interface, extending it to a script is quite natural and seamless. The automation of processing and analysis can easily be achieved.

The third category of image analysis and exploration tools consists of *image databases*. As the number and size of image data increase, the only way to effectively handle the data is through the use of database management systems. They typically allow us to organize the data, metadata, and analysis results in projects, and provide remote viewing capability of the images. They sometimes also support user annotations and allow us to launch batch analysis scripts on user-defined regions of interest. We will introduce several such systems.

The fourth category of tools consists of *libraries*. These tools are mainly interfaced for use by programming or scripting. Both GUI and CLI tools often use these libraries in their back end. As the direct use of image processing libraries is uncommon for the average biologist, we exclude them from this overview.

A summary of all tools appearing this chapter is provided in Appendix 2.A.

2.2

GUI Tools: Generic Platforms

2.2.1

ImageJ

ImageJ is one of the most widely used software for bioimage analysis [1]. It is a public domain software, completely open source and free to download (Table 2.1). It runs on the Java virtual machine, which allows the software to be largely free from dependence on operating systems. Usage is open to broad type of users, from biologists with marginal knowledge on image processing to

Table 2.1 Links to ImageJ packages.

Package	URL
ImageJ	http://imagej.nih.gov/ij/
Fiji	fiji.sc
ImageJ2	imagej.net/ImageJ2
Bio7	bio7.org

the experts of image processing algorithm development. It has an intuitive graphical user interface that allows easy access for beginners, but at the same time it allows serious development of plug-ins to add new functions. Many plug-ins are developed worldwide and most of them are freely offered. The community of ImageJ users and developers is huge: it has a highly active mailing list with nearly 1900 subscribers. The ImageJ website has had nearly 7000 visitors per day, and the full text search for "ImageJ" in Europe PubMed Central returned more than 55 000 articles (May 2015) and during 2014 there were 9280 articles using ImageJ. The last ImageJ conference in Luxembourg was attended by 119 people. The first release of ImageJ was in 1997, and as of May 2015, it is still under active development by Wayne Rasband. The flexible interface of ImageJ invoked a huge number of plug-in development projects, resulting in numerous implementations of image processing algorithms and convenient utilities. This availability of a variety of functions contributed largely to the spread of ImageJ all over the world. A problem associated with this wide impact was that dependencies between those plug-ins became complicated, and often hard to untangle. To resolve this, a distribution of ImageJ bundled with many plug-ins called Fiji was initiated in 2007 and has been actively maintained by a large group of developers [2]. In normal ImageJ there are about 500 commands, and in this distribution called Fiji, the number exceeds 900. The central feature of Fiji is its automatic plug-in updating function.

Bio7 is a unique distribution that focuses on merging ImageJ and R (see description in Section 2.7.4) in a single interface [3]. The passing of image analysis output to statistical analysis by R could be smoothly achieved.

Since 2009, ImageJ2 project has started to inherit the best part of ImageJ while upgrading the core architecture of the software. It is a complete rewrite of ImageJ, but its compatibility to ImageJ environment such as plug-ins and ImageJ Macro is maintained. ImageJ2 is already a part of Fiji, but by default Fiji runs on ImageJ. Switching to ImageJ2 back end is possible by changing the configuration.

2.2.2

Icy

Icy (icy.bioimageanalysis.org/) is a software platform for bioimage analysis with a strong emphasis on collaborative efforts [4]. Plug-ins for Icy are managed through its highly interactive website aimed at collecting inputs and feedbacks from both developers and biologists. Several different approaches are offered for constructing image analysis workflows. For biologists who are not accustomed to programming, a visual programming interface is provided that allows the user to graphically design image analysis workflows, just like LabVIEW and KNIME. Resulting visual programs are associated with the capacity to automatically check its plug-in dependencies and install any missing releases. This ensures the perfect reproduction of image analysis workflows documented in published papers. At the same time, JavaScript and Python are also integrated for scripting workflows in a conventional way. The learning curve for the beginner is

facilitated by rich examples and an intuitive keyword search. It integrates various Java computing libraries to ease the development of custom plug-ins. For example, ImageJ is included as a back end and ImageJ plug-ins can be launched from other plug-ins or from an integrated version of ImageJ. Other integrated libraries include ImgLib2, VTK, and OpenCV.

2.3

GUI Tools: Workflow Based

2.3.1

CellProfiler + CellProfiler Analyst

CellProfiler (www.cellprofiler.org/) is designed to enable biologists without training in computer vision or programming to quantitatively measure cell or whole-organism phenotypes from thousands of images automatically [5,6]. The researcher creates an analysis workflow called “pipeline” from “modules” that first find cells and cell compartments and then measure features of those cells to extract numerical data that characterize the biological objects and phenomena. Pipeline construction is structured so that the most general and successful methods and strategies are the ones that are automatically suggested, but the user can override these defaults and pull from many of the basic algorithms and techniques of image analysis to customize solution to problems.

Because it was initially developed for high-content screening image-based assay, it has been constructed to easily batch process thousands of images. User-defined pipelines can be saved and reused afterward.

CellProfiler pipeline is a simple text file and it could be executed from command line using CellProfiler.py script. CellProfiler can also be used as a library directly from Python script by importing modules. CellProfiler can be extended through plug-ins written in Python or for ImageJ.

CellProfiler Analyst is a companion software for CellProfiler, which can also be used by its own to analyze large data set of features extracted from images. It allows interactive exploration and analysis of measured data, including a supervised machine learning system to recognize subtle phenotypes.

2.3.2

ilastik

ilastik (ilastik.org/) is a framework, GUI, and suite of workflows that facilitate automated segmentation, classification, tracking, and counting in 2D and 3D multispectral images and videos [7]. These workflows are cast as interactive machine learning problems, which require no user experience in image processing but rather provide example annotations. Taking the pixel classification workflow as an example, ilastik has a convenient mouse interface for labeling an arbitrary number of classes in the images. These labels, along with a set of image

features, are then used for a machine learning-based method to classify image regions in different classes. In the interactive training mode, ilastik provides real-time feedback of the current classifier predictions and thus allows for targeted training and overall reduced labeling time. Once the classifier has been trained on a representative subset of the data, it can be used to automatically batch process a very large number of data sets. Other workflows follow similar usage steps. The plug-in functionality allows advanced users to add their own problem-specific features, apart from the provided set of features based on color, edges, and textures in the image. ilastik projects can be further imported to other tools such as CellProfiler and KNIME, for specific post-processing or analysis tasks.

2.3.3

Definiens Developer XD

Definiens Developer XD (developer.definiens.com/) is a commercial image segmentation and classification tool. The user designs a signal processing workflow by combining built-in filtering, thresholding, and object classification modules. Object detection is typically done on hierarchical object levels, for example, cell level for cell objects and organelle level for nucleus and ER objects inside a cell object. For each object, a huge set of features (shape-based, intensity-based, relations to neighboring objects, etc.) is available and can be used for object classification or merging with neighboring objects. The classical Definiens workflow is the so-called bottom-up approach: in the first step, the image is segmented in numerous small objects, resulting in a heavy oversegmentation of the target objects. Objects are then fused step by step on the basis of features such as the relative border to neighboring object or an elliptic fit of resulting (fused) object. Objects can be assigned to different classes (such as nucleus or cancer cell) based on their features.

2.4

GUI Tools: 4D + t Data Exploration and Analysis

2.4.1

Amira

Amira (now distributed and maintained by FEI, <http://www.fei.com/software/amira-3d-for-life-sciences/>) is a 3D visualization software, allowing also to do 3D processing and quantification (using Visilog and ITK libraries, as well as specific filters). Amira takes full advantage of the hardware acceleration and some filters could be run on NVidia graphical cores. The learning curve of Amira is eased by the presence of tutorials and example, but its object-oriented philosophy makes it sometimes difficult to apprehend for beginners, even though the functions are quite similar to professional GUI animation software. Amira can be controlled by the TCL scripting language, and is interfaced with Matlab. Additional modules can be created using C++. One of the main strength of

Amira is the number and quality of registration and optimization algorithm for automatic data fusion. It will be merged with Avizo (more physics/industry oriented) in the near future.

2.4.2

Arivis Vision4D

Arivis Vision4D (<http://vision.arivis.com/en/arivis-Vision4D>) is commercial software developed in Rostock, Germany. The main strength of Vision4D is its ability to handle really large data sets, for example, SPIM and EM data in the terabyte range. This is achieved through the use of a proprietary file format: When first opening and importing image data, the data are automatically copied into a sophisticated format that allows rapid Google Maps-like zooming and panning. It is possible to interactively visualize, process, segment, and analyze multidimensional data (3D + time + channels + scopes). The analysis is pipeline-based and several modules are preinstalled, for example, colocalization, FRET, and tracking. For this, filters/image processing steps from ITC and VIGRA are available. Import (even acquisition from TWAIN devices) of a wide, and growing, variety of image formats is supported and export in tiff, jpg, and so on is possible. Arivis WebView makes it possible to set up a web browser interface for collaborative work, for example, manual annotation of a large data set, allowing multiple users to simultaneously work on the same data. Arivis is a young company and its products are under rapid development at the moment of writing (2015).

2.4.3

Imaris

Bitplane Imaris (<http://www.bitplane.com/imaris/imaris>) is a commercial software for visualization and analysis of 3D data sets from fluorescence microscopy. The fast rendering engine together with the clear user interface makes it easy to reconstruct and interactively explore 3D data.

Image analysis tools are organized in workflows; that is, the user is guided through a wizard if he/she wants to detect and analyze objects in the image. One example is the "Find Spots" workflow, a method to segment spot-shaped objects and track them over time. Within the "Find Spots" wizard, the user is asked to define detection parameters step by step. Instant visual feedback allows us to fine-tune parameters as, for example, intensity thresholds. Besides others, workflows for segmenting cells as well as a neuron reconstruction tool are available. After completing the wizard, detected objects are rendered in 3D and can be overlaid with the original data. Statistics of detected objects can be exported to CSV files or visualized inside Imaris with the Vantage module. Due to the workflow-oriented approach, the software is very easy to use, but not very flexible. To overcome these restrictions, Imaris can be customized with plug-ins (so-called Xtensions) via the ImarisXT API. With the interface, image and object data can be accessed with scripts written in Matlab or Python.

Besides its image analysis capabilities, the software provides a very easy-to-use tool for rendering animations. The user defines some keyframe views by rotating and zooming in the data set and the software computes smooth camera movements from one keyframe to the next. The result can be exported to standard movie formats (e.g., avi format).

2.4.4

Volocity

Volocity (<http://cellularimaging.perkinelmer.com/downloads/>) is a commercial software package that was created by PerkinElmer. It consists of a free-of-charge core packet, called "Volocity LE," and several extensions with additional functionalities, which must be purchased: "Volocity LE" enables the user to import image sequences and perform basic image processing. The "Volocity Visualization" extension offers 3D rendering and interactive exploration of multichannel data sets. "Volocity Classification" allows for measuring and tracking of microscopy data in three dimensions over time. The "Volocity Restoration" package offers rapid and easy deconvolution, using measured or estimated point spread functions.

The advantages of Volocity lie in its ease of use. Volocity is designed to enable scientists without expertise in programming to visualize and analyze 3D-based data sets. A graphical user interface is used to process images in a "what you see is what you get" manner. Complex fluorescence z-stack images from wide-field and confocal microscopes are easily processed and analyzed.

While it is a powerful tool to read and prepare 3D fluorescence data for presentation and to process images, it does not allow the user to extend its functions by self-written macros. Therefore, this software is a tool for application only.

2.4.5

Vaa3D

Vaa3D (www.vaa3d.org) is a handy, fast, and versatile 3D/4D/5D image visualization and analysis open-source software for bioimages and surface objects [8]. It is particularly oriented toward filament-like structure tracing and provides some unique segmentation and analysis functions. It supports a very simple and powerful plug-in interface to extend its base functions.

2.5

GUI Tools: Image Restoration and Analysis

2.5.1

AutoQuant X

AutoQuant X (Media Cybernetics, www.mediacy.com/index.aspx?page=autoquant) is a commercial software suite mainly used for the deconvolution of fluorescence

microscopy images (wide-field, confocal, two-photon, and spinning disk) and their correction (aberration, registration, and bleaching). The software can also deal with image quantification (colocalization, ratiometry, and FRET) or data analysis and 3D visualization. In simple terms, deconvolution is an image restoration operation by which out-of-focus light is removed from a three-dimensional stack of images. The goal is to reconstruct the original emission pattern from the blurred and degraded version acquired with the microscope. The most advanced deconvolution algorithms provided (blind and nonblind maximum likelihood) are very robust, among the most efficient available, and their implementation is computationally highly optimized. Some more classical algorithms (no/nearest neighbors, inverse, and Wiener filters) as well as differential interference contrast (DIC) image restoration and 2D image deconvolution are also implemented.

2.5.2

SVI Huygens

SVI Huygens (svi.nl/HomePage) is a commercial software suite for the deconvolution of microscopy images. In simple terms, deconvolution is an image restoration operation by which out-of-focus light is attenuated from a 3D stack of images. The goal is to reconstruct the original emission pattern from the blurred and degraded version acquired with the microscope. The Huygens suite in its current version (version 15.05 at the time of writing) can be used with images acquired in the following modalities: wide-field, confocal, Nipkow confocal (spinning disk), STED, and SPIM. The software suite also includes optional visualization and analysis algorithms, including a surface renderer, a “simulated fluorescence process renderer,” object tracking, and colocalization, among others.

2.6

GUI Tools: Specialized Software

2.6.1

CellCognition

CellCognition (www.cellcognition.org/) is a computational framework dedicated to the automatic analysis of live cell imaging data in the context of high-content screening, initially specialized in the detection of mitotic events and to follow their time course [9]. Its GUI is called CeCogAnalyzer, which allows biologists to parameterize analysis workflow. It contains machine learning-based algorithms for segmentation of cells and cellular compartments based on various fluorescent markers, features to describe cellular morphology by both texture and shape, tools for visualizing and annotating the phenotypes, classification, tracking, and error correction. CellCognition can be used by novices in the field of image analysis and is applicable to hundreds of thousands of images in computer clusters with minimal effort.

2.6.2

NeuronStudio

NeuronStudio (<http://research.mssm.edu/cnic/tools-ns.html>) is an open-source software designed to allow reconstruction of neuronal structures from confocal and multiphoton images [10,11]. It is a self-contained software package that is free and easy to use. NeuronStudio provides tools for manual, semi-manual, and automatic tracing of the dendritic arbor as well as manual and automatic detection and classification of dendritic spines. In addition, advanced 2D and 3D visualization techniques facilitate the verification of the reconstruction, as well as allowing accurate manual editing. The output of the program is compatible with standard compartment modeling and morphometric software applications. The software is very optimized, but unfortunately still in beta release and the project is not active since 2009.

2.6.3

TMARKER

TMARKER (http://www.nexus.ethz.ch/equipment_tools/software/tmarker.html, <https://github.com/ETH-NEXUS/TMARKER>) is an open-source application for the detection and classification of nuclei in immunohistochemical (IHC) tissue microarrays (TMAs). The software is written in Java, is free, is easy to install, has a straightforward user interface, and comes with easy-to-follow tutorials and test data. TMARKER was developed for nuclei counting and nuclear IHC staining estimation of human cancer tissue and can be expected to perform best on similar data.

The main usage is the automated detection of nuclei in color images via color deconvolution and/or supervised via graph cuts or superpixels. After detection, nuclei can be classified via built-in supervised machine learning models using random forests, Bayesian networks, or support vector machine (SVM) algorithms. Some statistical diagnostics, such as *F*-score and precision/recall plots, can be performed directly in TMARKER. While technically advanced, the input from the user is usually limited to estimating the nucleus size and a few other parameters and then clicking on nuclei that exemplify the different classes of interest.

After segmentation and classification, the results can be exported as csv files, or as html to form a small report complete with results and images. Segmentation settings and results can also be saved as xml and tma (TMARKER) files and later loaded for further processing in TMARKER.

2.7

CLI Tools

With many of the tools discussed above, images are processed interactively while viewing the effect of processing on each image. In command line-based tools

listed below, images are typically imported without opening the image on desktop: they are loaded as an object in the memory and processing is done without visually checking changes to the image.

2.7.1

Matlab

Matlab (Matrix Laboratory, <http://www.mathworks.com/products/matlab/>) is a commercial integrated development environment (IDE) and a programming language oriented toward matrix manipulation and linear algebra launched by MathWorks in 1984. It is essentially a scripting language used to combine optimized high-level functions. To complement the extensive set of functions provided with the IDE, a large number of commercial and open toolboxes are available. The functions can be called from the interactive console or assembled as scripts. The most useful toolboxes for bioimage analysis are most certainly the image processing and the statistical toolboxes. Images are formally manipulated as matrices or higher dimension objects. Most common image processing and data analysis operations are implemented. Data visualization and simple GUI design are also extensively covered. Matlab is typically used for workflow prototyping, exchange, and publication. A large community of users is openly sharing their scripts. Matlab can also be used to release end products as the scripts can be compiled to executable files and ran outside the environment, or even encrypted. A free alternative to Matlab is Octave (<http://www.gnu.org/software/octave/index.html>), offering less toolboxes.

2.7.2

KNIME

KNIME (<https://tech.knime.org/community/image-processing>), the Konstanz Information Miner, is an open-source platform that contains modules for data integration, transformation, analysis, visualization, reporting, and integration [12]. KNIME's friendly graphical interface allows users to visually create workflows/pipelines as assembly of nodes. This enables the flexibility of independently executing one, some, or all nodes' tasks, and inspecting their results. KNIME is implemented in Java with extension mechanism. It also allows for wrappers calling other code written in Python, Perl. KNIME offers possibility to integrate other open-source projects such as R, LIBSVM, JFreeChart, ImageJ, CellProfiler, and the Chemistry Development Kit. Recently, ilastik has also provided projects that can be integrated with KNIME and CellProfiler developers are now collaborating together with KNIME team to allow KNIME to run CellProfiler pipeline. In terms of processing large data, it is only limited by the available hard disk space, as opposed to most other open-source tools that are limited to the available RAM.

The KNIME Image Processing plug-in is a large repository based on ImgLib2, a generic multidimensional image processing library also used in Fiji and Icy. In

particular, it provides >120 image formats, algorithms for preprocessing, filtering segmentation, feature extraction, tracking, and classification.

2.7.3

Python

Python (www.python.org/) is not a software package, but is a scripting language. The merit of Python is that there are numerous libraries for image processing and analysis. In terms of scripting, Python is more powerful than Matlab due to its bridging capability to many computer languages such as C, C++, and Java. Considering that the trend of image processing and analysis is getting more and more toward cross-language library usage, Python is a good choice to learn for high-end processing and analysis.

2.7.4

R/Image Processing in R

R is a software environment for statistical computing and graphics (Table 2.2). Numerous add-ons, called “packages,” written by researchers are freely available via the Comprehensive R Archive Network (CRAN). R is widely used in combination with other image processing software packages for analyzing numerical data extracted from image data and to plot those results. RStudio is an IDE for R. It combines many useful features such as markdown-based documentation system called “R Markdown” and the version control using Git. Besides its usefulness for statistical computing, image processing and analysis is also possible in R using “EImage” toolbox, a part of Bioconductor project.

2.7.5

LabVIEW

LabVIEW (Laboratory Virtual Instrument Engineering Workbench, <http://www.ni.com/labview/>) is a commercial visual programming language launched by National Instruments in 1986. The graphical language is named “G”. This graphical language naturally allows task parallelism and synchronization. It is primarily aimed at instrument control, and many devices (cameras, stages, lasers, etc.) come with LabVIEW drivers and software development kits. Like Matlab, LabVIEW became popular in many fields of engineering and its functions are organized in toolboxes. LabVIEW Vision is an extensive image processing and image analysis toolbox that

Table 2.2 Links to R and related packages.

Package	URL
R	www.r-project.org/
RStudio	http://www.rstudio.com/
EImage	http://www.bioconductor.org/packages/release/bioc/html/EImage.html

teams up well to drive hardware equipped with video camera and relying on computer vision for its operation (e.g., quality control, microscopes, etc.). It can, of course, also be used for independent, offline, image processing and analysis.

2.7.6

IGOR Pro

IGOR Pro (<https://www.wavemetrics.com/products/igorpro/igorpro.htm>) is a commercially available data analysis software developed by WaveMetrics, Inc. When it was introduced in 1989, it aimed at time series analysis. Since then, it has largely evolved and, nowadays, covers other applications such as curve fitting and image processing. A library for image processing is readily available, and an add-on for live image capturing is also available. Being mostly used by technical professionals, it is highly suitable for experimentation with scientific and engineering data and allows for the production of publication-quality graphs and page layouts. It comes with its own programming language called "IgorPro Procedure" and compiler allowing to extend the built-in functions or writing independent add-on in C or C++ ("XOP Tool Kit"). While this requires some analytical and programming skills, many users value the mixture of provided functions and freedom to extend them for a most suitable acquisition and analysis of data.

2.8

Image Databases

2.8.1

OMERO

OMERO (<http://www.openmicroscopy.org/site/products/omero>), developed by the Open Microscopy Environment consortium, is a software suite for the management of biological microscopy data. It is composed of a back-end server suite with the database and appropriate management software, and a set of front-end interfaces, which communicate at different levels with the server, such as desktop clients, web applications, and software libraries (to access the data remotely from Java, Matlab, Python, and other development platforms). Since version 5, the images are kept in their original format, which allows the user to have access to the original file, as well as to interact with the data through the front-end applications. It supports regions of interest, data tables, and the remote execution of analysis routines.

2.8.2

BisQue

BisQue (Bio-Image Semantic Query User Environment, bioimage.ucsb.edu/bisque) is a scientific image management tool organized around a server-client architecture. Its purpose is to store, visualize, organize, and analyze images in the cloud.

The images can be imported to the database (100+ biological formats supported) with relevant metadata and integrated to data sets. It is then possible to add hand-drawn annotations (possibly in collaboration) and to launch custom-made ImageJ, Matlab, or Python scripts to analyze the images. Search and comparison of data sets by image data and content are supported. The queries support high-level semantic articulations.

2.8.3

openBIS

The Open Source Biology Information System (openBIS, <http://www.cisd.ethz.ch/software/openBIS>) was developed by the Scientific IT Services (SIS) group at ETH Zurich, Switzerland. Its purpose is the management, annotation, and sharing of data. The openBIS software framework is extensible and has been customized for high-content screening, proteomics, metabolomics, and deep sequencing. Typically, openBIS is installed on a Linux server by a system administrator and accessed through a web interface by the users. openBIS is a free open-source software: powerful, well documented, and quite flexible. However, to take full advantage of this flexibility you do need to know a fair level of Python, shell scripting, and command line tools. Data can be stored in a distributed manner on multiple separate network-associated servers and the level of access via the web interface is easy to control, for example, which data a given user is allowed to read and/or edit. It is also possible to use openBIS as the platform when setting up a publication server where data are shared with the general public.

2.8.4

Avadis iMANAGE

Avadis iMANAGE (<http://www.strandls.com/solutions/strand-imanage>), developed in collaboration between Strand Life Sciences and Institut Curie imaging facility and IT department, is a ready-made commercial software suite for the management of biological microscopy data in particular on facilities. It provides shared, secure, uniform, and open access to image life cycle data and algorithms, through an easy access to cluster computing from the database web client. It has been thought to be highly scalable, and to deal with storage quota specifications and invoicing. Dynamic folder hierarchies can be created based on image and user metadata can be created on the fly. All data can be accessed and parsed by blocks (only relevant bits for a process can be chosen to be accessed). There is no conversion of data, neither duplication. A rich application programming interface in Java and SOAP allows easy extensions.

Appendix 2.A

Interfaces, execution environments, ecosystem, and licenses are summarized in Table 2.3.

Table 2.3 Summary of image analysis tools.

Name	GUI	CLI	Scripting	OS	Scripting Language	License
ImageJ	O	O	O	All	ImageJ Macro, Javascript, Jython, JRuby, BeanShell, Groovy, Clojure	Public domain
Icy	O	O	O	All	ImageJ Macro, Javascript, Jython, Protocol (Graphical Programming)	GPL v3
CellProfiler/Analyst	O	O	O	All	Python, Workflow	GPL v2
ilastik	O	O	O	All	Python	GPL
Definiens Developer XD	O	O	O	All	Definiens Rulesets (Graphical Programming)	Commercial
Amira	O	O	O	All	TCL, Python	Commercial
Arivis Vision4D	O	X	O	Win	Python	Commercial
Imaris	O	X	X	Win, OSX	Matlab	Commercial
Volocity	O	X	X	Win, OSX	—	Commercial
Vaa3D	O	O	X	All	—	MIT
AutoQuant X	O	X	X	Win	—	Commercial
Huygens	O	O	O	All	TCL	Commercial
CellCognition	O	O	O	All	Python	LGPL
NeuronStudio	O	X	X	Win	—	Original
Matlab	X	O	O	All	Matlab	Commercial
KNIME	O	O	O	All	Workflow (Graphical Programming), Perl, Python, ImageJ Macro, Groovy (Java), R, Matlab	GPL
Python	X	O	O	All	Python	Python Software Foundation License
R	X	O	O	All	R	GPLv2/v3
LabVIEW	X	O	O	All	LabVIEW VI (Graphical Programming)	Commercial
IGOR Pro	X	O	O	Win, OSX	Igor Procedure	Commercial
OMERO	O	O	O	All	Python	GNU public "copyleft" license
BisQue	O	X	X	All	—	BSD (modified)
openBIS	O	O	O	All (client)/Linux (server)	Java, Jython	Apache Software License 2.0

(continued)

Table 2.3 (Continued)

Name	GUI	CLI	Scripting	OS	Scripting Language	License
Avadis iMANAGE	O	X	O	All	SOAP, Matlab, ImageJ macro, ICY Protocol	Commercial
TMARKER	O	X	X	All	—	GPL v2

References

- Schneider, C.A., Rasband, W.S., and Eliceiri, K.W. (2012) NIH Image to ImageJ: 25 years of image analysis. *Nat. Methods*, **9** (7), 671–675.
- 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., and Cardona, A. (2012) Fiji: an open-source platform for biological-image analysis. *Nat. Methods*, **9** (7), 676–682.
- Austenfeld, M. and Beyschlag, W. (2012) A graphical user interface for R in a rich client platform for ecological modeling. *J. Stat. Software*, **49** (4), 1–19.
- de Chaumont, F., Dallongeville, S., Chenouard, N., Hervé, N., Pop, S., Provoost, T., Meas-Yedid, V., Pankajakshan, P., Lecomte, T., Le Montagner, Y., Lagache, T., Dufour, A., and Olivo-Marin, J.C. (2012) Icy: an open bioimage informatics platform for extended reproducible research. *Nat. Methods*, **9** (7), 690–696.
- Carpenter, A.E., Jones, T.R., Lamprecht, M.R., Clarke, C., Kang, I.H., Friman, O., Guertin, D.A., Chang, J.H., Lindquist, R.A., Moffat, J., Golland, P., and Sabatini, D.M. (2006.) CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biol.*, **7** (10), R100.
- Kamentsky, L., Jones, T.R., Fraser, A., Bray, M.A., Logan, D.J., Madden, K.L., Ljosa, V., Rueden, C., Eliceiri, K.W., and Carpenter, A.E. (2011) Improved structure, function and compatibility for CellProfiler: modular high-throughput image analysis software. *Bioinformatics*, **27** (8), 1179–1180
- Sommer, C., Straehle, C., Kothe, U., and Hamprecht, F.A. (2011) ilastik: interactive learning and segmentation toolkit. Proceedings of the International Symposium on Biomedical Imaging, pp. 230–233.
- Peng, H., Ruan, Z., Long, F., Simpson, J.H., and Myers, E.W. (2010) V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets. *Nat. Biotechnol.*, **28** (4), 348–353.
- Held, M., Schmitz, M.H.A., Fischer, B., Walter, T., Neumann, B., Olma, M.H., Peter, M., Ellenberg, J., and Gerlich, D.W. (2010) CellCognition: time-resolved phenotype annotation in high-throughput live cell imaging. *Nat. Methods*, **7** (9), 747–754.
- Rodriguez, A., Ehlenberger, D.B., Dickstein, D.L., Hof, P.R., and Wearne, S.L. (2008) Automated three-dimensional detection and shape classification of dendritic spines from fluorescence microscopy images. *PLoS One*, **3** (4), e1997.
- Wearne, S.L., Rodriguez, A., Ehlenberger, D.B., Rocher, A.B., Henderson, S.C., and Hof, P.R. (2005) New techniques for imaging, digitization and analysis of three-dimensional neural morphology on multiple scales. *Neuroscience*, **136** (3), 661–680.
- Berthold, M., Cebon, N., Dill, F., Gabriel, T., Kötter, T., Meinel, T., Ohl, P., Sieb, C., Thiel, K., and Wiswedel, B. (2008) KNIME: The Konstanz Information Miner, in *Data Analysis, Machine Learning and Applications*, Studies in Classification, Data, Analysis, and Knowledge Organization (eds C. Preisach, H. Burkhardt, L. Schmidt-Thieme, and R. Decker), Springer, Berlin, pp. 319–326.

