Improvement, structure, function and compatibility for CellProfiler: modular high-throughput image analysis software

Lee Kamentsky1, Thouis R. Jones1, Adam Fraser1, Mark-Anthony Bray1, David J. Logan1, Katherine L. Madden1, Vebjorn Ljosa1, Curtis Rueden2, Kevin W. Eliceiri2 and Anne E. Carpenter1,

1Imaging Platform, Broad Institute of Harvard and MIT, Cambridge, MA 02142 and 2Laboratory for Optical and Computational Instrumentation, University of Wisconsin, Madison, WI 53706, USA

ABSTRACT
Summary: There is a strong and growing need in the biology research community for accurate, automated image analysis. Here, we describe CellProfiler 2.0, which has been engineered to meet the needs of its growing user base. It is more robust and user friendly, with new algorithms and features to facilitate high-throughput work. ImageJ plugins can now be run within a CellProfiler pipeline.

Availability and Implementation: CellProfiler 2.0 is free and open source, available at http://www.cellprofiler.org under the GPL v. 2 license. It is available as a packaged application for Macintosh OS X and Microsoft Windows and can be compiled for Linux.

Contact: anne@broadinstitute.org

Supplementary information: Supplementary data are available at Bioinformatics online.

Received on November 2, 2010; revised on January 31, 2011; accepted on February 14, 2011

1 INTRODUCTION

CellProfiler is freely available, open-source software that enables researchers without training in computer programming to measure biological phenotypes quantitatively and automatically from thousands of images. With an interface designed by biologists and underlying algorithms developed by computer scientists, CellProfiler bridges the gap between advanced image analysis algorithms and scientists who lack computational expertise. In the 4 years since its publication (Carpenter et al., 2006; Lamprecht et al., 2007), it has been rapidly adopted by the worldwide biological community and cited in more than 250 articles. Roughly half of its users are outside the USA. CellProfiler was initially designed for high-throughput image analysis but is often used for small-scale projects. This highlights the trend toward quantifying information in images regardless of experiment size.

CellProfiler’s interface lets researchers build customized chains of interoperable image analysis modules to identify and measure biological objects and features in images. These modular pipelines can be saved and shared with colleagues. CellProfiler has been used to measure individual cells, colonies of cells and whole organisms in a wide range of assays (e.g. counting cells, measuring staining intensities and scoring complex phenotypes with machine learning) and at many experimental scales (from a few to hundreds of thousands of images). A variety of cell types have been analyzed, including budding yeast, Drosophila, mouse, rat and dozens of human cell types. The diverse measurements generated by CellProfiler provide raw material for machine-learning algorithms that can identify challenging phenotypes (Jones et al., 2009; Misselwitz et al., 2010; Ramo et al., 2009).

CellProfiler fills a unique role in the software landscape. It is a modular, high-throughput, open-source biological image analysis package, and it won the 2009 Bio-IT World Best Practices Award in IT & Informatics. CellProfiler 2.0 improves upon the design of the original version, resulting in professionally engineered software with improved usability and functionality, as well as integration with other open-source image-related software.

2 IMPROVEMENTS IN CELLPROFILER 2.0

Robust infrastructure and interoperability: we redesigned the software’s infrastructure while porting it from the proprietary MATLAB language to the open-source Python language, making use of the high-performance scientific libraries NumPy and SciPy (Oliphant, 2007). While retaining the successful attributes of CellProfiler 1.0 (Supplementary Fig. 1 and Table 1), CellProfiler 2.0 compares favorably to CellProfiler 1.0 in terms of performance (Supplementary Fig. 2) and features (Supplementary Table S2).

Object-oriented design and professional software practices were integral to the porting effort, including version control, a continuous build process and the development of an extensive validation suite.

CellProfiler 2.0 is designed to be extensible and interoperable; its plug-in interface allows outside developers to write and distribute new CellProfiler modules. We use Cython (http://www.cython.org) to implement computationally intensive algorithms, as well as bridge to precompiled libraries including Java via the Java Native Interface. The Java/Python bridge allows CellProfiler 2.0 to load nearly 100 image formats via the Open Microscopy Environment Consortium’s Bio-formats library (http://www.loci.wisc.edu/software/bio-formats). Because 5% of CellProfiler-citing papers also used ImageJ (http://rsweb.nih.gov/ij), we built a bridge to run ImageJ macros in the context of a CellProfiler pipeline. In our own research, we have used third-party ImageJ plug-ins via CellProfiler to enhance neurites in images (Supplementary Fig. 1A) and detect focal planes in three-dimensional images.
FlagImage module can exclude images from analysis based on or SQLite databases during image processing. CellProfiler 2.0’s database capabilities and is now able to upload directly to MySQL images can also be loaded similarly. CellProfiler 2.0 has enhanced microscopes or laboratory information systems. Metadata about the image file locations, which might be generated by automated via HTTP or located based on a comma-delimited text file containing instances running on separate computing cores or cluster nodes in a in batch mode: sets of images are partitioned between CellProfiler (Supplementary Table 3).

of Gabor (Supplementary Fig. 3) and Haralick texture features shape features (Supplementary Fig. 2) and improved measurement (Knutsson and Westin, 1993), faster measurement of Zernike-based three classes instead of the typical two. Other changes include an algorithm for more accurate operations on masked images (Knutsson and Westin, 1993), faster measurement of Zernike-based shape features (Supplementary Fig. 2) and improved measurement of Gabor (Supplementary Fig. 3) and Haralick texture features (Supplementary Table 3).

Enhancements for high-throughput use: CellProfiler 2.0 can be run in batch mode: sets of images are partitioned between CellProfiler instances running on separate computing cores or cluster nodes in a distributed environment. In CellProfiler 2.0, images can be loaded via HTTP or located based on a comma-delimited text file containing image file locations, which might be generated by automated microscopes or laboratory information systems. Metadata about the images can also be loaded similarly. CellProfiler 2.0 has enhanced database capabilities and is now able to upload directly to MySQL, or SQLite databases during image processing. CellProfiler 2.0’s FlagImage module can exclude images from analysis based on measurements of image quality, such as bluriness and presence of debris. Images can be grouped for aggregate operations, such as illumination correction of images on a per-plate basis or analysis of multiple time-lapse movies or three-dimensional image stacks. More detailed information on CellProfiler and high-throughput screening is available at http://www.cellprofiler.org/hcs.html.

Future directions: we will use the improved infrastructure and design of CellProfiler 2.0 as the basis for our future work. Where feasible, we will continue to leverage existing open-source projects to add functionality, such as software for workflow management (e.g. OMERO and KNIME) and classification of pixels or whole images by machine learning (e.g. Windchrm and Ilastik). While supporting contributions from other developers, we will also develop novel algorithms for CellProfiler based on our ongoing research, including time-lapse and three-dimensional image analysis, metrics and corrections for assay quality control and performance evaluation and algorithms for Caeorhabditis elegans image-based screens (Riklin-Raviv et al., 2010; Walthby et al., 2010).

ACKNOWLEDGEMENTS

The authors thank the members of their laboratories for contributing to the development of the software and this article, especially Shrvas Rao and Emily Schloff.

Funding: National Institutes of Health (R01 GM089652-01 to A.E.C., RC2 GM092519-01 to K.W.E. and NIH RL1 HG004671, which is administratively linked to RL1 CA133834, RL1 GM084437 and UL1 RR024924).

Conflict of Interest: none declared.

REFERENCES
