Bioinformatics, 37(21), 2021, 3992–3994
doi: 10.1093/bioinformatics/btab634
Advance Access Publication Date: 3 September 2021
Applications Note

Bioimage informatics

CellProfiler Analyst 3.0: accessible data exploration and machine learning for image analysis

David R. Stirling, Anne E. Carpenter and Beth A. Cimini *

Imaging Platform, Broad Institute of MIT and Harvard, Cambridge, MA 02142, USA

*To whom correspondence should be addressed.

Associate Editor: Jinbo Xu
Received on July 20, 2021; revised on August 27, 2021; editorial decision on August 30, 2021

Abstract

Summary: Image-based experiments can yield many thousands of individual measurements describing each object of interest, such as cells in microscopy screens. CellProfiler Analyst is a free, open-source software package designed for the exploration of quantitative image-derived data and the training of machine learning classifiers with an intuitive user interface. We have now released CellProfiler Analyst 3.0, which in addition to enhanced performance adds support for neural network classifiers, identifying rare object subsets, and direct transfer of objects of interest from visualization tools into the Classifier tool for use as training data. This release also increases interoperability with the recently released CellProfiler 4, making it easier for users to detect and measure particular classes of objects in their analyses.

Availability: CellProfiler Analyst binaries for Windows and MacOS are freely available for download at https://cellprofileranalyst.org/. Source code is implemented in Python 3 and is available at https://github.com/CellProfiler/CellProfiler-Analyst/. A sample dataset is available at https://cellprofileranalyst.org/examples, based on images freely available from the Broad Bioimage Benchmark Collection.

Contact: bcimini@broadinstitute.org

1 Background

With the increasing adoption of high-throughput microscopy, scientists have been able to generate large datasets containing thousands of individual images. This necessitates automated computational analysis to efficiently derive biological insights from the raw data. Free software packages such as ImageJ (Schneider *et al.*, 2012) and CellProfiler (McQuin *et al.*, 2018) allow users to extract hundreds or thousands of numerical measurements from their image data, but it is ultimately on the user to determine which of these features are relevant to the biological problem being investigated. Spreadsheet programs are familiar but lack features and integration with source images that biologists often need. CellProfiler Analyst is a data exploration package for helping users to explore and extract information from large datasets, including (but not limited to) those produced by CellProfiler pipelines (Jones *et al.*, 2008). The software includes several tools for users to visualize and filter their datasets, alongside tools for training machine learning classifier models within a convenient graphical user interface that is geared toward working with image data (Dao *et al.*, 2016). While other tools such as Ilastik (Berg *et al.*, 2019) and Advanced Cell Classifier (Piccinini *et al.*, 2017) can provide a GUI for training classifiers with image-based data, these do not include data exploration and visualization tools like those in CellProfiler Analyst. These tools provide an intuitive interface for scientists to explore their data in forms such as histograms and scatter plots, though without the advanced statistical tools or extreme customizability of a pure programming language. Herein we present CellProfiler Analyst 3.0, which includes major performance improvements and new features that improve the utility of the software.

2 General changes

We ported CellProfiler Analyst to the Python 3 programming language to ensure compatibility with future operating systems after the official Python 2 end-of-life in 2020. We also revised the program's builds to package all Java dependencies within the main installer, which dramatically simplifies the installation process. In addition, we added a faster imageio-based image loader to supplement the existing bioformats-based implementation (Silvester *et al.*, 2020). While bioformats provides broader file format compatibility, imageio allows for common image formats to be loaded more efficiently, which greatly improves the time to generate object thumbnails. At present neither of these loaders supports accessing only partial sections of an image, which may be an area for further development to handle large files produced by whole slide imaging.

CellProfiler Analyst 3.0 can export machine learning models that are compatible with CellProfiler 4.2+, allowing the resulting classifiers to be directly embedded into CellProfiler pipelines. This maintains and expands upon the interoperability between the two programs.

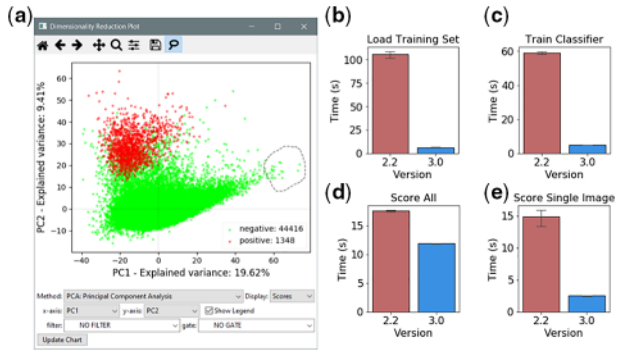

Fig. 1. Improvements in CellProfiler Analyst 3.0. **a**) Screenshot of the new Dimensionality Reduction tool, using principal component analysis with the example dataset. Marker colors represent classification results from a previously generated classifier. The lasso tool has been activated to select objects of interest, which can then be used to train a classifier. **b**) Comparison of execution time between CPA 2.2 and CPA 3.0 when loading a training set of 600 objects in 13 classes. **c**) Execution time to run the training sequence for a FastGentleBoosting classifier. **d**) Time taken to score 45 000 objects using a FastGentleBoosting classifier. **e**) Execution time to score a single image containing 130 objects, then annotate results onto a preview image.

3 New features

We added a dimensionality reduction tool to help users visualize the variance within high-dimensional datasets (Fig. 1A). This is particularly important when creating classifiers that can identify rare images in a set; rare outliers are needed for training but may be difficult to find through random sampling. When the dataset contains thousands of measurements users can struggle to identify those which reveal these outliers. Dimensionality reduction allows the numerous measurements generated by CellProfiler to be condensed into a smaller subset of features which represent the overall variance in the dataset. This provides a more manageable series of features which the user can then explore.

This tool supports multiple common reduction techniques including principal component analysis (Pearson 1901), Feature Agglomeration (Jain *et al.*, 1999) and t-Distributed Stochastic Neighbor Embedding (van der Maaten and Hinton 2008). The resulting components can be visualized on a scatter plot, which can color individual objects by class if the classification was previously performed on the dataset. As with other tools, users can create gates to isolate specific populations for further analysis.

We improved the gating functionality so that gates drawn in plotting tools are immediately made available within the Classifier tool, rather than having to be manually converted into filters. The dimensionality reduction tool contains a new lasso tool for selecting objects of interest through drawing a custom polygon. Right-click menu options allow the selected objects to be sent directly to an open Classifier window for use as training data. These improvements simplify the process of finding populations of interest and creating training sets for machine learning models in the Classifier.

In the Classifier tool we added support for automatic rescaling of input data, based on the scikit-learn StandardScaler implementation (Pedregosa *et al.*, 2011). This improves the performance of certain classifier types (such as K-Neighbors) that are skewed by the absolute values of measurements. It also now supports tunable neural network classifiers based on the scikit-learn MLPClassifier class (Hinton 1989), which can be useful for performing complex nonlinear classification tasks. New keyboard shortcuts for sorting objects into classification bins provide a more rapid means of generating training sets than dragging-and-dropping.

4 Performance improvements

We improved performance in several areas of the Classifier tool. We optimized loading of previously saved training sets by batching objects during database fetching, reducing the loading time for a sample dataset by over 10-fold (Fig. 1B). We also found that removing redundant database calls and unnecessary caching steps during classifier model training reduced processing time by 90% (Fig. 1C).

Database handling improvements were also made in the functions for scoring the datasets after training. These refinements produced a modest improvement in the performance of the 'score all' (Fig. 1D) function. We further optimized the 'score image' function by revising the workflow for fetching object coordinates, which significantly reduced the time taken to display a scoring preview overlay (Fig. 1E). Usability improvements to the image viewer include a shortcut to resize the image to fit the window and a table display outlining counts of each class found in the image.

5 Future directions

The next major frontier for phenotype classification is to train deep learning models straight from raw pixels rather than requiring a separate feature extraction step (Lucas *et al.*, 2021). Enabling scoring of phenotypes that fall along a continuum rather than into discrete bins would also be useful, as we recently found in red blood cell aging (Doan *et al.*, 2020). A remaining limitation in our software is that only MySQL and SQLite databases can be accessed, though migrating toward using packages such as sqlalchemy could expand on compatibility in the future. Ultimately, maintaining CellProfiler Analyst as an up-to-date resource for users to explore high-dimensional, image-based data without needing to code will help the biology community for years to come.

Acknowledgements

The authors would like to thank Pearl Ryder, Erin Weisbart, Jane Hung, David Dao, Egor Zindy and Mario Emmenlauer for contributions to bug fixes and testing of pre-release versions of this software. They also thank all the members of the bioimaging community who have provided feedback and suggestions which have helped to guide this work.

Funding

This work was supported by the National Institutes of Health grants (R35 GM122547 and P41 GM135019 to A.E.C.). This project has been made possible in part by grant number 2020-225720 to B.A.C. from the Chan Zuckerberg Initiative DAF, an advised fund of the Silicon Valley Community Foundation. The funders had no role in study design, data collection and analysis, decision to publish, or manuscript preparation.

Conflict of Interest: none declared.

References

Berg,S. *et al.* (2019) Ilastik: interactive machine learning for (bio)image analysis. *Nat. Methods*, **16**, 1226–1232.

Dao,D. *et al.* (2016) CellProfiler analyst: interactive data exploration, analysis and classification of large biological image sets. *Bioinformatics*, **32**, 3210–3212.

Doan,M. *et al.* (2020) Objective assessment of stored blood quality by deep learning. *Proc. Natl. Acad. Sci. USA*, **117**, 21381–21390.

Hinton,G.E. (1989) Connectionist learning procedures. *Artif. Intell.*, **40**, 185–234.

Jain,A.K. *et al.* (1999) Data clustering. *ACM Comput. Surv.*, **31**, 264–323.

Jones,T.R. *et al.* (2008) CellProfiler analyst: data exploration and analysis software for complex image-based screens. *BMC Bioinformatics*, **9**, 482.

Lucas,A.M. *et al.* (2021) Open-source deep-learning software for bioimage segmentation. *Mol. Biol. Cell*, **32**, 823–829.

McQuin,C. *et al.* (2018) CellProfiler 3.0: next-generation image processing for biology. *PLoS Biol.*, **16**, e2005970.

Pearson,K. (1901) LIII. On lines and planes of closest fit to systems of points in space. *Lond Edinburgh Dublin Philos. Mag. J. Sci.*, **2**, 559–572.

Pedregosa, F. *et al*. (2011). *Scikit-learn: machine learning in Python*. *JMLR*, 12, 2825–2830.

Piccinini,F. *et al*. (2017) Advanced cell classifier: user-friendly machine-learning-based software for discovering phenotypes in high-content imaging data. *Cell Syst*., **4**, 651–655.e5.

Schneider,C.A. *et al*. (2012) NIH image to ImageJ: 25 years of image analysis. *Nat. Methods*, **9**, 671–675.

Silvester,S. *et al*. 2020. Imageio/imageio, July. doi:10.5281/zenodo.4972048.

van der Maaten,L. and Hinton,G. (2008) Visualizing data using T-SNE. *J. Mach. Learn. Res*., **9**, 2579–2605.