CellProfiler Analyst Web (CPAW) - Exploration, analysis, and classification of biological images on the web

Bella Baidak*
University of Massachusetts Boston
Yahiya Hussain†
University of Massachusetts Boston
Emma Kelminson‡
University of Massachusetts Boston
Thouis R. Jones§
The Broad Institute of MIT and Harvard
Loraine Franke¶
University of Massachusetts Boston
Daniel Haehn‖
University of Massachusetts Boston

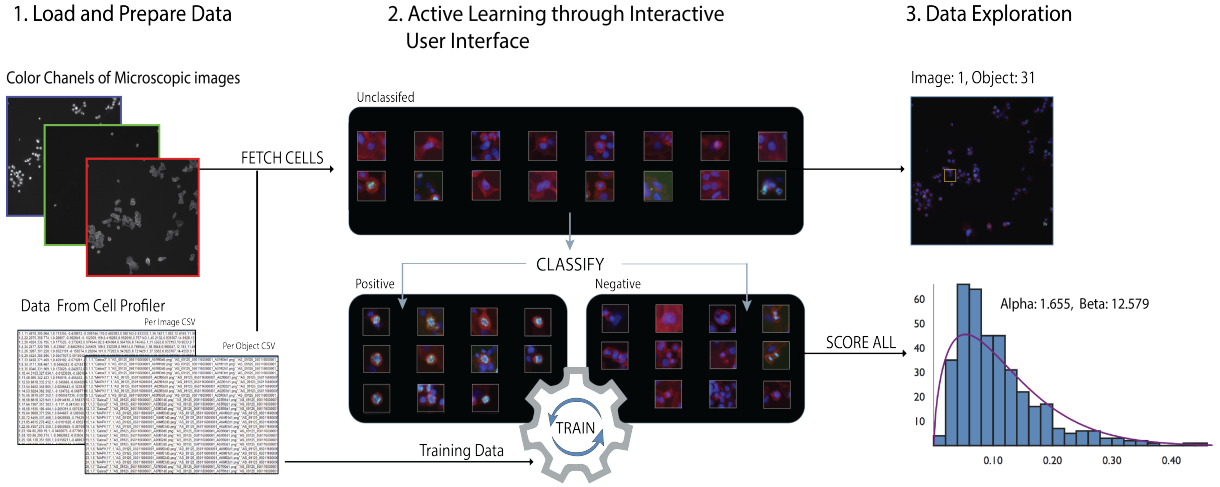

Figure 1: Workflow through CellProfiler Analyst Web: The first step shows the data received from CellProfiler (left) with biological microscopy images of cells, then fetching cells to be trained in the second active learning step with tensorflow.js and classified by the machine learning classifier into its corresponding class. The third step shows the final scores ready for data exploration and analysis by the user. A demo of CellProfiler Analyst Web can be found here: `https://mpsych.github.io/CellProfilerAnalystWeb/`.

Abstract

CellProfiler Analyst (CPA) has enabled the scientific research community to explore image-based data and classify complex biological phenotypes through an interactive user interface since its release in 2008. This paper describes CellProfiler Analyst Web (CPAW), a newly redesigned and web-based version of the software, allowing for greater accessibility, quicker setup, and facilitating a simple workflow for users. Installation and managing new versions has been challenging and time-consuming, historically. CPAW is an alternative that ensures installation and future updates are not a hassle to the user. CPAW ports the core iteration loop of CPA to a pure server-less browser environment using modern web-development technologies, allowing computationally heavy activities, like machine learning, to occur without freezing the user interface (UI). With a setup as simple as navigating to a website, CPAW presents a clean UI to the user to refine their classifier and explore phenotypic data easily. We evaluated both the old and the new version of the software in an extensive domain expert study. We found that users could complete the essential classification tasks in CPAW and CPA 3.0 with the same efficiency. Additionally, users completed the tasks 20 percent faster using CPAW compared to CPA 3.0. The code of CellProfiler Analyst Web is open-source and available at `https://mpsych.github.io/CellProfilerAnalystWeb/`.

Index Terms: Biological Images—Visualization—Machine Learning Classification—Evaluation Methods;

1 Introduction

Analyzing microscopy cell images is a significant challenge in biology, where unstructured images of cells must be converted to structured data of visual cellular phenotypes. The analysis and exploration of large imaging datasets is an important area in biological research, of which flexible software for image analysis is a critical component. CellProfiler Analyst (CPA) is software designed for biologists and data scientists to explore, visualize, and classify biological image-based data through an interactive user interface. Using data from CellProfiler [12], a feature extraction software for images, CPA can help identify complex and subtle phenotypes, improve quality control, and provide single-cell and population-level information from experiments. The core features of the current version of CPA are its ability to handle millions of cell images, drag and drop user interface for phenotypic classification, object-centric scoring of groups of cells, and data visualization tools.

CellProfiler Analyst has proven to be a helpful tool for the scientific community; unfortunately, it has remained a desktop application, requiring time-consuming and complicated installation and upgrades, including dependency management. Historically CellProfiler Analyst has been prone to software instabilities and update lags, causing difficulty to existing users and new users. Though

*e-mail: bella.baidak001@umb.edu
†e-mail: yahiya.hussain001@umb.edu
‡e-mail: emma.kelminson001@umb.edu
§e-mail: thouis@broadinstitute.org
¶e-mail: franke@mpsych.org
‖e-mail: daniel.haehn@umb.edu

CellProfiler Analyst 3.0 [18] has improved the installation processes, including updating the original codebase to modern Python, working with CPA 3.0 still requires the user's commitment to installation, and it continues to be susceptible to future complications around updating versions.

Here, we introduce CellProfiler Analyst Web (CPAW), an easy-to-use open-source web-based version of the software with an improved user experience. CellProfiler Analyst Web provides a clean user interface using modern programming tools and architecture while maintaining CPA's core cell visualization and machine learning capabilities. It removes user management of system libraries, installation difficulties and invisibly provides updates, and is easily made available to users via loading a web page. We compare the application version CPA 3.0 versus the new fully web-based version CellProfiler Analyst Web through an expert user study. Four domain experts at the home institution of CellProfiler Analyst completed timed tasks in either CPA or CPAW. Our results show that experts could complete core classification tasks with similar efficiency and 20 percent less time with CPAW than CPA 3.0.

2 Related Work

CellProfiler [4] was initially released in 2005 and enabled the scientific research community to create flexible image analysis pipelines. The most recent version, CellProfiler 3.0, was published in 2018, including enhancements for image processing of 3D image stacks [13]. Based on CellProfiler Analyst 2.0 [5,11] we developed a novel web-based version that allows users to explore, classify and analyze cellular phenotypes in biological datasets interactively. Open-source software tools are an important component in many research fields, and biological-image analysis is no exception. Other systems for analyzing biological image data sets include Imagej2 [17] and KNIME [6], Advanced Cell Classifier [16], and Ilastik [2]. The latter differs from the previous packages in leveraging machine learning to create more flexible image-analysis workflows. Orbit Image Analysis [7] takes in data pipelines from CellProfiler and shares some machine learning techniques, object/cell segmentation, and object classification features with CellProfiler Analyst. QuPath [1] offers users an automatic cells detection and random trees classification tool for brightfield and fluorescent images. Another tool, CellCognition [9], uses image processing and computer vision for tasks such as single-cell tracking and classification of cell morphologies. Moreover, the downloadable desktop-app WND-CHARM WND-CHARM [15] is not only beneficial for biological or cell images but has a multi-purpose image classifier. However, these software tools all require a local download, powerful machines, or long processing times. To the best of our knowledge, there is no other work using an entirely client-side web application for biological image processing and machine learning in the browser.

3 Design Objectives and Tasks

Visual analysis of cell samples has dramatically impacted and furthered modern biology. New methods have given biologists the capability to capture image data from cells treated with libraries of potential drugs or gene-perturbing reagents, such as CRISPR-based activation or inactivation of gene expression. Automated microscopes then collect hundreds of thousands of images of treated cells. CellProfiler, CellProfiler Analyst, and other tools described above have given biologists easy access to image-based information. Allowing them to analyze and interpret patterns and relationships within data, identify quality-control issues, and make serendipitous discoveries in their image-based screens.

A typical use-case scenario for the tool requires the user to load a dataset previously processed by CellProfiler to identify and measure individual cells. Upon loading image data, the user can use simple drag-and-drop actions to quickly add labeled cells to the training set according to phenotype, through a pseudo-active-learning loop [5]. At the end of a training session, the user can export the classifier and training set. This allows for later refinement of the training sets or the application of the classifier to enormous data sets via an offline system. In addition, the user can score all images in the entire experiment and search for images where the phenotype is enriched or de-enriched relative to other images, corresponding to treatments that affect the phenotype.

CellProfiler Analyst Web currently lacks some of the additional data exploration features of CPA; however, these features are not needed for cell classification. Experts reported in interviews that most time spent using CPA was with the classification tools.

3.1 Tasks

In close collaboration with domain experts, we derived a list of tasks reflecting the typical step-by-step workflow researchers and biologists perform working with microscopy cell images in CPA:
T1 - *Load data from CellProfiler into CellProfiler Analyst.* This initial step gives CellProfiler Analyst the information necessary to display vibrant cell images for the user to explore and classify.
T2 - *Fetch cells at random, classify, train, and evaluate.* When starting the classification process, the user fetches cells randomly from the full population, then drags and drops the cell images into positive or negative bins based on the corresponding phenotype. When the user has labeled a few dozen examples of each class, they train the classifier. After training, the user can view a confusion matrix to evaluate the classifier's accuracy.
T3 - *Fetch positive and/or negative cells, classify, train, and evaluate.* After labeling a sufficient number of cells and training the classifier, a user repeatedly fetches cells the classifier identifies as positive or negative, adding misclassified cells to the correct training set. After training, they can again evaluate the state of the classifier by viewing the confusion matrix.
T4 -*Fetch confusing cells, classify, train, and evaluate.* The user can fetch cells confusing to the classifier and then complete the same classifying, training, and evaluating process to improve the classifier's performance. Here, "confusing" cells are cells that are not obviously one class or another according to the current classifier; classifying them is analogous to active learning in other systems [5].
T5 - *Score all images and explore the data.* When the user is satisfied with the accuracy of the classifier, the user can score all the data. Scoring all gives the user information about the number of positive and negative cells in each image or group of images. At any point, the user can double click on an individual cell image to see the cell in the context of the whole image.
T6 - *Save the training set and classification model.* When users feel they have finished classifying and exploring the image data, they can save the classification model specifications and training set for later use or future refinement.

3.2 Visualization requirements and choices

As seen in Figure 2, CellProfiler Web user interface maintains the original structure of CellProfiler Analyst while simplifying the design and providing a more intuitive experience to the user. CPAW displays classification actions in buttons, making the typical workflow apparent to the user. We made the design decision to use action nomenclature for the buttons so the user can easily apply familiarity with CPA to CPAW, allowing them to use CPAW quickly and intuitively the first time they encounter it [3]. During training, CPAW will show a real-time plot with epochs, loss, and accuracy to the user. Furthermore, being web-based, CPAW can be used on nearly any internet-enabled PC.

In an initial pre-design user study, we conducted discussions with multiple domain experts from the Broad Institute of MIT and Harvard (the home institution of CellProfiler and CellProfiler Analyst development) to assess the requirements and design objectives for CPAW. We found the following requirements for the visualization

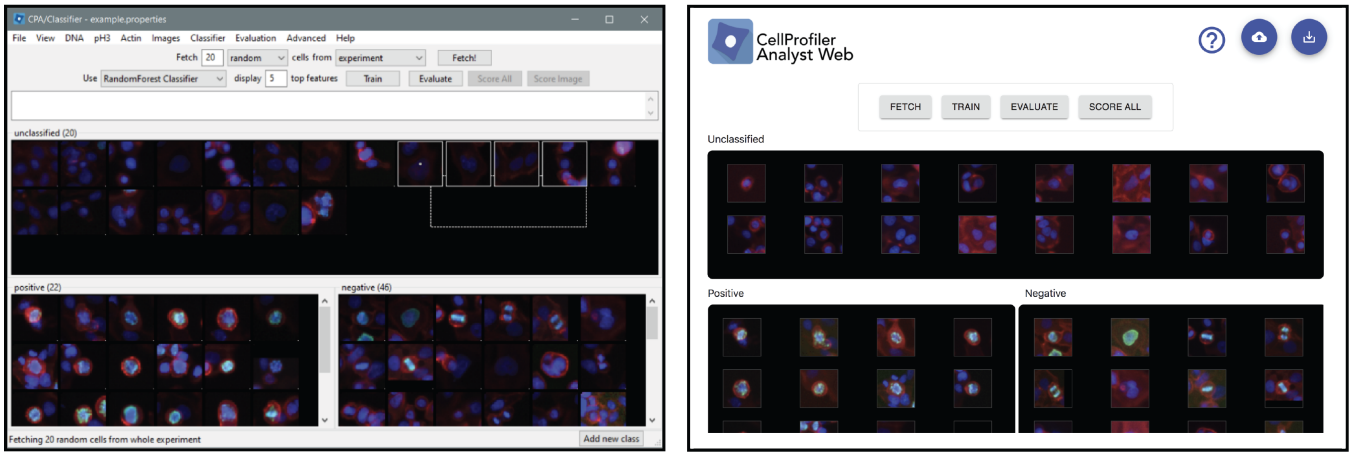

Figure 2: CellProfiler Analyst 3.0 user interface on the left and CellProfiler Analyst Web user interface on the right side. React.js and Material.ui provide a clean and modern designed user interface look in the browser. CellProfiler Analyst Web is accessible without the need of any installation, while CellProfiler Analyst 3.0 requires installation which can be difficult for some users.

tool to be critical for our domain experts:
R1 - The software should be entirely web-based and only rely on client-side computation, requiring no back-end.
R2 - The software must have similar classification functionality as the current version of CPA, including fetching cells by phenotype, training a classifier with modern machine learning techniques, and scoring the entire data set with a classifier.
R3 - The software must utilize an easy-to-use drag-and-drop user interface, allowing the user to quickly train and correct a classifier by dropping cells in phenotype bins.
R4 - The software must be able to parse CellProfiler's output files as input for backward compatibility.
R5 - The software must allow the user to download the finished classifier's specification and parameters and the training set that they can later use to explore the data further or refine later.
R6 - The software must be interactive and reactive to user-input independent of background computations and processes.
R7 - The software has to be as performant in its machine learning pipeline and codebase as CPA.

Thus, the main objective of CPAW was to emulate the essential classification features of CPA while prioritizing usability and accessibility. In addition, our aim in creating a more intuitive design was to speed up the classification process and give users a more satisfying experience.

4 Implementation Details

The UI of CPAW was implemented using React.js, a library developed by Facebook for creating websites with dynamic user interfaces with JavaScript ECMA 6 (**R1**). The UI was also supplemented with Bootstrap and Material UI libraries to create a pleasing user interface design. TensorFlow.js was used for machine learning (**R2, R5, R7**) and React-grid-dnd for the drag-and-drop UI (**R3, R6**). We decided to use the React framework for this project over the vanilla JavaScript approach because of React's usage of JSX components in a virtual DOM environment, which allows the programmer to quickly and effectively break up a monolithic HTML codebase into dynamic, intuitive, and reusable components.

There are various competitors to React with other component-based web-development frameworks like Angular.js and Vue.js. However, for us, React stood above the rest for its combination of a gentle learning curve, thorough documentation, lightweight programming boilerplate, and its large community of extensions. React.js also has a useful one-way data-binding design, which facilitates hierarchical, structured programs, useful for scalability and readability.

These choices of React and various extensions fulfill **R1**, ensuring that CPAW is entirely web-based and only relies on the client-side. We wrote code to parse CellProfiler's output files using Paraparse for parsing CSV files (**R4**), and the browser for loading images, and TensorFlow for running active ML operations and subsequent data analysis. However, such heavy computation caused freezing and stuttering in the single-thread browser environment of JavaScript. To ameliorate this, we made heavy use of WebWorkers and IndexedDB to ensure desktop-app-like interactivity by moving expensive computations to different logical cores of the client's machine (**R5, R6**). Though this did involve adding some complexity to the code for inter-worker communication.

To score the entire dataset, we classify every cell, count each cell's phenotype in every image, and apply Empirical Bayes shrinkage estimates [10] to calculate a per-image estimate of the enrichment of the classified phenotypes. To do so, we fit a Beta-Binomial distribution to the per-image phenotype counts and extract the parameters of the Beta(α, β) component. We then adjust the ratio of per-class to total cells in each image with a shrinkage estimate, using this equation:

$$\frac{phenotype + \alpha}{total + \alpha + \beta} \tag{1}$$

Where "phenotype" and "total" are the number of cells of a given phenotype and the total number of cells in an individual image, respectively. This process gives a robust estimate of the phenotype fraction in each image, even in images with few cells.

The implementation described above fulfills **R1-R7**, giving CPAW have the same functionality for classification as the current version of CPA in a web-based, highly interactive, client-side system.

5 Expert User Study

We interviewed four domain experts using CellProfiler Analyst for their research, two men and two women, through semi-structured interviews of around 30 minutes via an individual Zoom meeting. All experts were experienced in using the CellProfiler Analyst classifier function and have worked at the Broad Institute for about one year to three or more years. The expert's use of CPA ranged from just a few projects to monthly use to weekly use. To avoid bias during the user study, we randomly assigned two experts to work with CellProfiler Analyst 3.0 and the other two experts to use CellProfiler Analyst Web. We derive the following hypotheses:

Table 1: Average time (in seconds) domain experts needed to complete tasks for each tool. Improvement of time (in %) of CPAW vs. CPA 3.0. Tasks 2-4 consisted of multiple sub-tasks with fetching, classifying, training and evaluating, which required more time than other tasks.

Task	CPA 3.0	CPA Web	Improvement
Task 1	**18.565 s**	43.845 s	-136.17 %
Task 2	102.61 s	**70.625 s**	31.171 %
Task 3	119.93 s	**74.485 s**	37.892 %
Task 4	150.555 s	**97.88 s**	34.987 %
Task 5	**29.455** s	45.2 s	-53.454 %
Task 6	9.175 s	**8.495 s**	7.411 %
Total	430.29 s	**340.53 s**	20.86 %

H1: Both CPAW and CPA 3.0 can perform the same tasks and the process workflow equally well (following the requirement **R2**).
H2: Performing the process workflow with CPAW is faster than with CPA 3.0 (following the requirement **R7**).

At first, we introduced CPAW to the experts and gave them a brief overview of its functionality. We asked the experts to complete tasks that demonstrated the core classification functionality of CellProfiler Analyst. The detailed list of 16 sub-tasks was derived from the six main tasks ***T1 - T6*** we identified in Section 3.1. Before working on these tasks, we allowed the experts to familiarize themselves with the example data set they would use in the study and its particular cellular phenotype (not timed). After working with the example data set, we acquired more information concerning the overall workload during the study tasks with the NASA task load questionnaire [8]. Additionally, we recorded insightful qualitative feedback concerning the visualization and the interface design, such as other remarks and comments to the software.

Data Set. The data set we used for our expert study can be found on `https://cellprofileranalyst.org/examples`. This data set included a single 96-well plate of images from HT29 colon cancer cells. This cell line has been widely used for the study of many normal and neoplastic processes. The images were produced using the Cellomics ArrayScan instrument at the Whitehead–MIT Bioimaging Center, where each image has a size of 512x512 pixels. These images were a subset of those initially created for a lentiviral RNAi library targeting human and mouse genes, applied to an arrayed viral high-content screen [14]. The subset of data we used was 352.3 MB in size.

Results. The study results of the time consumed for each task are summarized in Table 1. The results indicate that overall, CPAW is as performant as CPA (**R7**), demonstrated through the expert's ability to complete all essential classification tasks using both tools (***H1***, ***R2***). On average, it took users a shorter amount of time to complete their assigned tasks using CPAW than with CPA 3.0, 5.67 minutes vs. 7.17 minutes on average (***H2***, ***R7***). We found CPAW to improve the average duration of the tasks by 20.86% compared to CPA 3.0. Especially Task 2 and Task 4, where the users had to fetch cells, train the network, and evaluate the classification, CPAW outperformed CPA 3.0. Tasks 2-4 required more time because the user had to complete the sub-tasks of fetching, classifying, training, and evaluating; these tasks require more thought and consideration. Our results from the NASA-TLX show CPA has an average score of 3.0, and CPAW has an average score of 2.9. These results indicate that CPA and CPAW are comparable tools requiring the same low mental effort for experts to use.

Qualitative Feedback. Before building CPAW and the updates to CPA 3.0, we interviewed one of our experts to learn more about the user's pain points. One particular comment on CPA 2.0 was: *"It's so buggy, that it's annoying"*. When asked how they work around current problems in CPA, they shared *"I have been trained by super users, who know the tricks and ways to get around what is missing"* and *"I am with the people that wrote the software"*. This statement is concerning for any user that does not work at the Broad Institute or closely with the developers of CPA, especially since one of its stated goals is to be accessible and useful for all biologists that wish to use it [5, 11]. Furthermore, we received constructive feedback and further feature suggestions after having the experts use CellProfiler Analyst Web. Expert's suggestions included adding a color display and intensity adjustments for cell images and selecting multiple images at once when dragging and dropping abilities.

Overall, the expert users were enthusiastic about the web version of the software, especially when considering new users and researchers using CPA for short-term projects in the future. One of the experts stated, *"the biggest appeal is having it easier for new users to use"*, and that *"installing CPA was sometimes not easy."*. Another expert expressed that *"it's definitely convenient to use if you don't want to download and install CPA itself"*.

6 Discussion and Conclusion

We presented CellProfiler Analyst Web, a fully web-based version of CellProfiler Analyst. While we do not aim to replace CellProfiler Analyst 3.0, our expert study proves that CellProfiler Analyst Web can be used as an alternative or parallel to CPA. CPAW offers many advantages of modern web applications, including ubiquitous accessibility with only an internet connection. It requires no installation and no difficulties for the user around updates, and is the only tool available entirely online, to our knowledge. A limitation of the software is that only recent versions of the Edge and Chrome browser are currently supported due to cutting-edge APIs in the program. Another limitation is that CellProfiler Analyst Web currently stores everything in RAM, making its data memory limits dependent on the browser and computer limits.

We are currently working on extending data handling capacities to larger data sets, adding a fine-tuned deep-learning model, easing RAM requirements, and potentially increasing performance. Further future work will also ensure that CPAW can be run smoothly in all commonly used modern browsers. Additionally, as per the expert suggestions, we will continue to add features to improve the user's experience and functionality of CPAW, including adding color and intensity adjustments for cell images and selecting multiple images at once when dragging and dropping. We also intend to add a well-plate diagram (present in CPA 3.0) and additional machine learning classification options.

As scientific software becomes more complex and specific, its usability can suffer. Scientists using these technologies could better exert effort and time performing their research instead of learning a given tool's specific setup and interface. While some software complexities are unavoidable, web-based tools allow for portability, minimal setup, and an extensive selection of easily accessible UI designs. In this paper, we have presented a tool, CellProfiler Analyst Web, a reimplementation of a phenotypic cell image analysis tool that was previously desktop-only. Though it lacks some features of CellProfiler Analyst, CPAW shows that powerful scientific tools can be implemented and deployed as an accessible web app. With continued development, CPAW will allow scientists in the biological community to continue to focus on their craft without a steep learning curve.

Acknowledgments

The authors wish to thank the domain experts for participating in the user study. Thank you to Emmanuel Adeniyi, Jing Yuan, and Corey O'Connor for your contributions in the early stages.

References

[1] P. Bankhead, M. B. Loughrey, J. A. Fernández, et al. Qupath: Open source software for digital pathology image analysis. *Scientific reports*, 7(1):1–7, 2017.

[2] S. Berg, D. Kutra, T. Kroeger, et al. Ilastik: interactive machine learning for (bio) image analysis. *Nature Methods*, 16(12):1226–1232, 2019.

[3] A. Blackler, V. Popovic, and D. Mahar. Intuitive interaction applied to interface design. *Proceedings International Design Congress - IASDR*, pp. 1–10, 01 2005.

[4] A. E. Carpenter, T. R. Jones, M. R. Lamprecht, et al. Cellprofiler: image analysis software for identifying and quantifying cell phenotypes. *Genome biology*, 7(10):1–11, 2006.

[5] D. Dao, A. N. Fraser, J. Hung, et al. Cellprofiler analyst: interactive data exploration, analysis and classification of large biological image sets. *Bioinformatics*, 32(20):3210–3212, 2016.

[6] A. Fillbrunn, C. Dietz, J. Pfeuffer, et al. Knime for reproducible cross-domain analysis of life science data. *Journal of biotechnology*, 261:149–156, 2017.

[7] I. G. Goldberg, C. Allan, J.-M. Burel, et al. The open microscopy environment (ome) data model and xml file: open tools for informatics and quantitative analysis in biological imaging. *Genome biology*, 6(5):1–13, 2005.

[8] S. G. Hart and L. E. Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Advances in psychology*, vol. 52, pp. 139–183. Elsevier, 1988.

[9] M. Held, M. H. Schmitz, B. Fischer, et al. Cellcognition: time-resolved phenotype annotation in high-throughput live cell imaging. *Nature methods*, 7(9):747–754, 2010.

[10] T. R. Jones, A. E. Carpenter, M. R. Lamprecht, et al. Scoring diverse cellular morphologies in image-based screens with iterative feedback and machine learning. *Proceedings of the National Academy of Sciences*, 106(6):1826–1831, 2009. doi: 10.1073/pnas.0808843106

[11] T. R. Jones, I. H. Kang, D. B. Wheeler, et al. Cellprofiler analyst: data exploration and analysis software for complex image-based screens. *Bioinformatics*, 9(1):1–16, 2008.

[12] L. Kamentsky, T. R. Jones, A. Fraser, et al. Improved structure, function and compatibility for CellProfiler: modular high-throughput image analysis software. *Bioinformatics*, 27(8):1179–1180, 02 2011.

[13] C. McQuin, A. Goodman, V. Chernyshev, et al. Cellprofiler 3.0: Next-generation image processing for biology. *PLoS biology*, 16(7):e2005970, 2018.

[14] J. Moffat, D. A. Grueneberg, X. Yang, et al. A lentiviral rnai library for human and mouse genes applied to an arrayed viral high-content screen. *Cell*, 124(6):1283–1298, 2006. doi: 10.1016/j.cell.2006.01.040

[15] N. Orlov, L. Shamir, T. Macura, et al. Wnd-charm: Multi-purpose image classification using compound image transforms. *Pattern recognition letters*, 29(11):1684–1693, 2008.

[16] F. Piccinini, T. Balassa, A. Szkalisity, et al. Advanced cell classifier: user-friendly machine-learning-based software for discovering phenotypes in high-content imaging data. *Cell systems*, 4(6):651–655, 2017.

[17] C. T. Rueden, J. Schindelin, M. C. Hiner, et al. Imagej2: Imagej for the next generation of scientific image data. *BMC bioinformatics*, 18(1):1–26, 2017.

[18] D. R. Stirling. Cellprofiler analyst 3.0 release: Accessible data exploration and machine learning for image analysis., Apr 2021. `https://carpenterlab.broadinstitute.org/blog/cellprofiler-analyst-3-0-release`.