https://doi.org/10.1038/s41592-025-02942-6

NimbusImage: a cloud-computing platform for image analysis

Imaging has been critical to biological discoveries for centuries. Throughout, one of the primary challenges has been the quantitative analysis of these images. The solutions to date have largely involved custom software in combination with software packages such as ImageJ[1], napari[2], CellProfiler[3], MATLAB, and others. However, there remains a need for users to interact with their data even if they lack the ability to code. New machine-learning algorithms have the potential to scale our ability to accurately quantify imaging data, but the technical expertise required to deploy these tools puts them out of reach for many users.

Here, we introduce NimbusImage, a software package that addresses these challenges. NimbusImage brings advancements in image analysis to users who may otherwise find such tools difficult to use, all in an easy-to-use web-based platform. Key features include cloud-based deployment, an intuitive interface that enables direct interaction with data, an extensible API (application programming interface), and plug-ins that combine conventional analytical methods with newer deep-learning techniques (Fig. 1).

Cloud-based computing refers to the deployment of software through servers instead of running locally. It has numerous advantages in deployment, scalability, and sharing, but it can present challenges in delivering a performant experience through a web browser. NimbusImage consists of a back end that handles computations and a front end that handles the user interface. The back end is powered by Girder (https://github.com/girder/girder), an open-source content management system developed by Kitware, and its large_image plug-in (https://github.com/girder/large_image), which enables Girder to read, process, and serve image datasets at scale. The large_image plug-in can read a large number of image formats and enables performant image navigation.

Girder also supports computation through self-contained 'workers'. These workers can batch-process large amounts of data in the background, allowing users to continue working with their data asynchronously. Workers have independent environments, enabling interoperability of packages with conflicting dependencies. The front end uses the Vue framework for a user-friendly interface, and front end–back end interactions have been heavily optimized to ensure fast performance. NimbusImage can be installed on servers for use by a group or run locally. Girder provides a file system for flexible data organization – in particular, grouping datasets into 'collections' that can be analyzed together in a consistent manner. Server deployment also facilitates easy data sharing without large file transfers.

Another challenge with most image-analysis pipelines is directly interacting with intermediate elements of the analysis. A key element of most analyses is the annotation of particular features within an image. Most annotation is either done manually (for example, by circling cells) or via automated pipelines, using algorithms such as Cellpose[4] or StarDist[5] to generate many annotations. Most users desire a middle ground that combines the power of automated analysis tools with the ability to manually review and fix the inevitable errors that these tools make, such as removing inappropriate 'cells' generated in autofluorescent regions. NimbusImage enables such interactions by separating image annotation from annotation measurement, using tags as a flexible system for organizing annotations and analyses. A typical workflow might have a user run Cellpose to generate cell segmentations tagged with 'cell'. Then, the user would identify incorrect cell annotations, remove them and manually circle missed cells. Finally, properties such as area, fluorescence intensity, and so on can be computed. These analyses combine automated and manual review seamlessly. Tagging also allows complex analyses, such as counting all transcription sites within nuclei while ignoring mature transcripts.

Machine learning has transformed major areas of image analysis in recent years. NimbusImage integrates machine learning analyses through a user-friendly interface. For example, we have implemented the cell segmentation tool Cellpose and our own spot-detection algorithm, Piscis[6]. Both were implemented using our containerized annotation worker API. The performance of machine learning tools typically slows as image size increases. We therefore developed the DeepTile framework (https://github.com/arjunrajlaboratory/deeptile), which splits the image into smaller tiles, processes each separately or in parallel, and then reassembles the annotations post facto, allowing users to perform analyses on arbitrarily large images and developers to focus on algorithms. NimbusImage's interface also enables human-in-the-loop retraining and fine-tuning of models such as Cellpose and Piscis based on human annotations. NimbusImage also incorporates the Segment Anything Model (SAM)[7] from Meta. This segmentation model enables the user to either click or draw a box on the image to identify objects with virtually real-time feedback, even when no specialized algorithm exists for a particular use case.

NimbusImage enables measurements of objects called 'properties'. These include morphometric analyses, such as area and perimeter, and various intensity measurements for points and blobs. 'Connections', which connect a parent annotation to a child annotation, allow, for instance, annotations to be connected over time in a time-lapse dataset or across *z* levels in a *z*-stack.

NimbusImage workers are built using an API for accessing data and annotations. The API can also be accessed programmatically via Python from a Jupyter notebook, enabling programmatic control. All annotations and properties can be exported and imported, ensuring full documentation and transparency of the analysis.

A problem with publishing imaging data is documenting so-called representative images, often criticized for their lack of transparency and traceability. NimbusImage provides 'Snapshots', which act as bookmarks in the dataset. A snapshot is an image area with contrast and channel information. Snapshots can be downloaded as images or movies that can be inserted into publications, including scale bars and annotations. This system dramatically enhances the traceability of images.

To enhance accessibility, NimbusImage includes a chatbot with knowledge of most

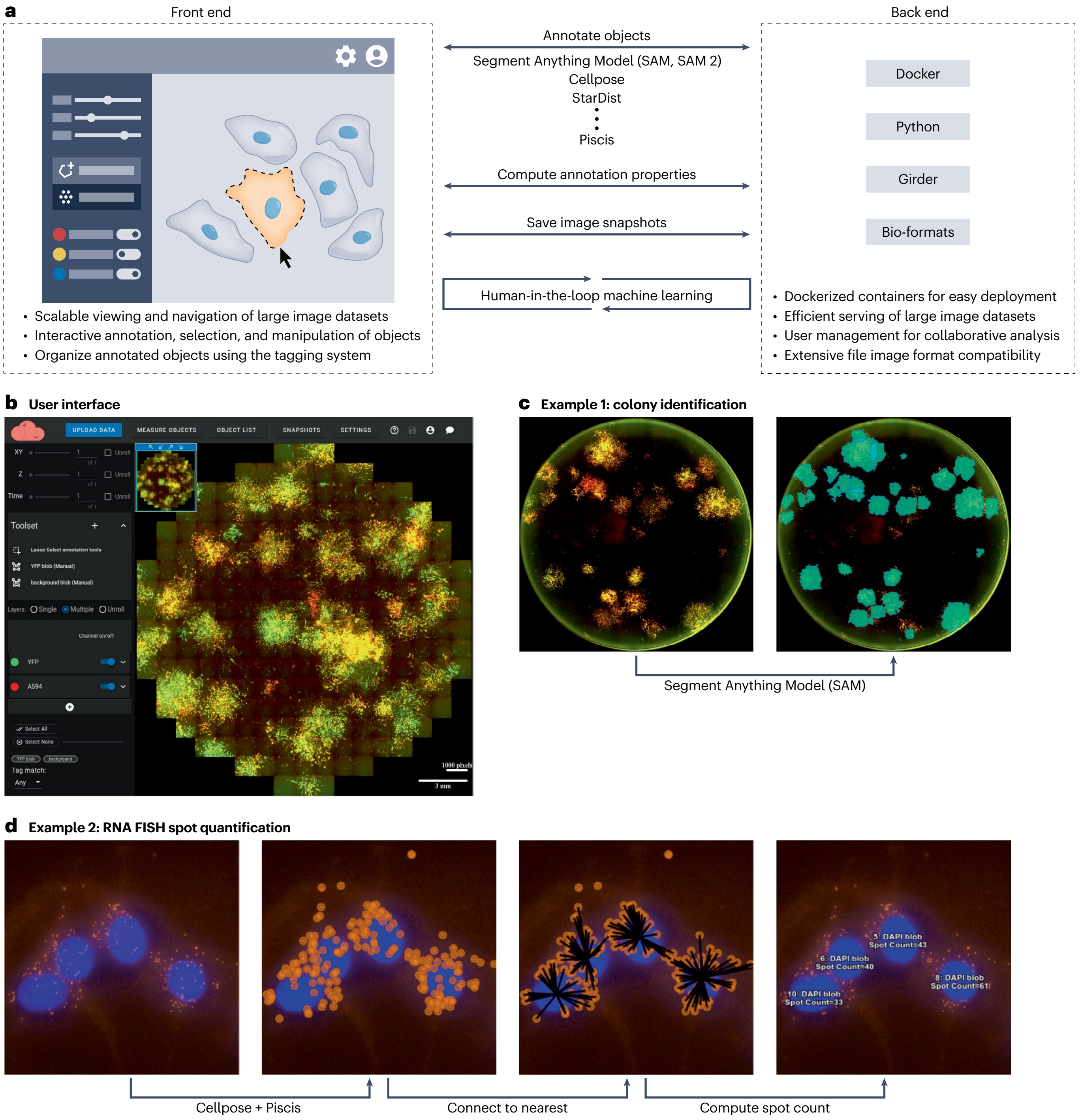

Fig. 1 | NimbusImage is a cloud-computing platform for image analysis. **a**, NimbusImage has a front-end interface in the browser that interfaces with a powerful back end to handle large images. **b**, User interface for a large stitched fluorescence microscopy image with multidimensional navigation and user-specified annotation tools. Additional panels allow users to view a list of all annotated objects, measure their properties, and save image snapshots for inclusion in publications. **c**, Example workflow for colony identification using SAM. **d**, Example workflow for RNA fluorescence in situ hybridization (RNA FISH) spot quantification, in which Cellpose is used for cell segmentation and Piscis for spot detection. Workers then connect spots to the nearest nucleus (up to a specified distance threshold) and count the number of assigned spots per cell.

features in the package. The chatbot can 'see' the user's images and suggest analyses and workflows for many tasks based on the user's research goals.

We believe that NimbusImage will help to democratize image analysis by allowing users to interact flexibly with their data while still being able to take advantage of advancements in machine learning.

Code availability

A commercial deployment of NimbusImage is available at https://nimbusimage.com/ and is feature-matched with the open-source version. Complementary documentation and tutorials are available at https://arjun-raj-lab.gitbook.io/nimbusimage. The source code for NimbusImage can be found in the GitHub repository https://github.com/arjunrajlaboratory/NimbusImage. Our curated collection of workers can be found at https://github.com/arjunrajlaboratory/ImageAnalysisProject. A.R. is the founder of CytoPixel Software, which develops and maintains the commercial version of the NimbusImage platform; however, all source code for the NimbusImage platform will remain open source.

Zijian Niu[1,2,3], Thibault Bruyère [4], David Manthey [5], Jingxin Li[6], Aoife O'Farrell[7] & Arjun Raj [7,8] ✉

[1]Department of Chemistry, School of Arts and Sciences, University of Pennsylvania, Philadelphia, PA, USA. [2]Department of Physics and Astronomy, School of Arts and Sciences, University of Pennsylvania, Philadelphia, PA, USA. [3]Computational and Systems Biology Program, Massachusetts Institute of Technology, Cambridge, MA, USA. [4]Kitware SAS, Villeurbanne, France. [5]Kitware Incorporated, Clifton Park, NY, USA. [6]Genetics and Epigenetics, Cell and Molecular Biology Graduate Group, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA, USA. [7]Department of Bioengineering, School of Engineering and Applied Sciences, University of Pennsylvania, Philadelphia, PA, USA. [8]Department of Genetics, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA, USA.

✉e-mail: arjunraj@seas.upenn.edu

References

1. Rueden, C. T. et al. *BMC Bioinformatics* **18**, 529 (2017).
2. Chiu, C.-L., Clack, N. & the Napari Community. *Microsc. Microanal.* **28**, 1576–1577 (2022).
3. Stirling, D. R. et al. *BMC Bioinformatics* **22**, 433 (2021).
4. Stringer, C., Wang, T., Michaelos, M. & Pachitariu, M. *Nat. Methods* **18**, 100–106 (2021).
5. Weigert, M., Schmidt, U., Haase, R., Sugawara, K. & Myers, G. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* 3666–3673 (2020).
6. Niu, Z. et al. *Cell Systems* https://doi.org/10.1016/j.cels.2025.101448 (in the press).
7. Ravi, N. et al. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2408.00714 (2024).

Acknowledgements

The authors thank members of the A.R. laboratory, particularly P. Burnham and L. Beck, for insightful discussions and feedback related to this work. The authors also thank A. Boucaud, L. Macron, and P. Choisel for their development efforts. A.R. acknowledges support from the Penn Epigenetics Institute. Z.N. acknowledges support from the Roy and Diana Vagelos Scholars Program in the Molecular Life Sciences, Roy and Diana Vagelos Science Challenge Award, Barry M. Goldwater Scholarship, Fannie and John Hertz Foundation Fellowship, Paul & Daisy Soros Fellowship for New Americans, and US Department of Energy Computational Science Graduate Fellowship (Department of Energy, Office of Science, and Office of Advanced Scientific Computing Research under award no. DE-SC0025528). J.L. acknowledges support from US National Institutes of Health Medical Scientist Training Program T32GM007170. A.O. acknowledges support from US National Science Foundation GRFP DGE-2236662.

Author contributions

A.R. conceived the project. Z.N. and A.R. implemented automatic workers to create annotations and connections and compute annotation properties. Z.N. designed the DeepTile library and integrated it into annotation workers. T.B. built the front-end software. D.M. built the back-end software. J. L. and A.O. tested the user interface and workers and identified issues by analyzing real imaging datasets.

Competing interests

A.R. receives royalties related to Stellaris RNA FISH probes and also is the founder of CytoPixel Software, which maintains a commercial version of NimbusImage.

Additional information

Peer review information *Nature Methods* thanks Timothy Monko and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Primary Handling Editor: Rita Strack, in collaboration with the *Nature Methods* team.