ImJoy: an open-source computational platform for the deep learning era

To the Editor — Deep learning (DL) methods achieve breakthrough performances in analyzing biomedical data across countless tasks, including medical diagnostics, DNA sequence analysis, augmented microscopy and drug design. Combined with increasing data repositories in genomics, imaging and other fields, such successes underlay a growing demand to adapt DL methods to new datasets and questions. However, the dissemination of DL approaches faces considerable hurdles. Most published DL studies require users to retrain models on their own data to obtain the best performance and/or avoid erroneous results. Although trained models are frequently available through web applications or ImageJ plugins, retraining is typically only possible via scripts or command lines, rather than graphical user interfaces (GUIs). In addition, the complexities of setting up the required hardware and software environments often constitute forbidding obstacles. Furthermore, the large datasets and computational resources typical of current DL successes pose challenges to traditional desktop-oriented software that tightly couple GUI and computation. Cloud services can partly alleviate these difficulties, but raise privacy and confidentiality issues that can be prohibitive for medical data. Meanwhile, deploying scientific software to mobile platforms can make them accessible to billions of people, enabling large-scale biomedical research and citizen science. These opportunities and challenges call for new computational frameworks.

Recognizing this need, we developed ImJoy (https://imjoy.io), an open-source plugin platform designed to deliver advanced, yet easy-to-use data analysis tools, especially based on DL (Fig. 1a, Supplementary Note 1). ImJoy’s core is a serverless progressive web application offering a fast and reliable user experience across all mainstream platforms, including laptops and mobile devices. ImJoy’s functionalities are provided by independently operating plugins that can be organized into workflows. Plugins written in different programming languages, including Javascript and Python, can communicate and exchange data bidirectionally through transparent function calls (Fig. 1a). Importantly, computation is decoupled from the GUI and can be performed either in the browser, or distributed locally or remotely through ‘plugin engines’ running on a workstation, remote servers or cloud services. Despite running in a browser, ImJoy works without server connection; once plugins have been downloaded from the Internet, together with other required resources (for example, specific software libraries or pre-trained DL models), they are securely isolated, sandboxed and thus suitable to process sensitive data. Imjoy makes it easy to build applications using existing software libraries, integrate with established platforms, such as Jupyter and ImageJ or Fiji, and process data on various computing infrastructures (Fig. 1a, Supplementary Note 2). Imjoy provides a built-in code editor for developing and testing plugins. A plugin’s source code is a single file that can be hosted on GitHub and installed with a single, easily shareable hyperlink (Fig. 1b). Plugin repositories can be made easily accessible by a gallery view (Fig. 1c). Next, we highlight ImJoy’s main capabilities, along with example plugins (Supplementary Note 2). For detailed documentation, including tutorials on porting existing tools to ImJoy, see https://imjoy.io/docs.

By leveraging various web libraries, such as three.js and D3, ImJoy allows users to build rich and interactive applications (Fig. 1a). For example, the ImageAnnotator plugin allows annotation of images, a pre-requisite for training segmentation methods, and can run on touchscreen devices (for example, smartphones). The HPA-UMAP plugin visualizes protein localization features computed by DL from Human Protein Atlas images. Such interactive visualizations are instrumental to exploring massive databases and interacting with cloud services.
To accelerate the adoption, improvement and testing of DL methods and promote reproducible research in biomedicine and beyond.

Code Availability
Source code for ImJoy and the example plugins is available at https://imjoy.io and https://github.com/imjoy-team/example-plugins, respectively.

Wei Ouyang1,2*, Florian Mueller*, Martin Hjelmare3, Emma Lundberg3,4 and Christophe Zimmer1

1 Imaging and Modeling Unit, Institut Pasteur, UMR 3691 CNRS, C3BI USR 3756 IP CNRS, Paris, France. 2 Science for Life Laboratory, School of Engineering Sciences in Chemistry, Biotechnology and Health, KTH - Royal Institute of Technology, Stockholm, Sweden. 3 Department of Genetics, Stanford University, Stanford, CA, USA. 4 Chan Zuckerberg Biohub, San Francisco, CA, USA.

*Correspondence: weio@kth.se; muellerf.research@gmail.com

Published online: 28 November 2019
https://doi.org/10.1038/s41592-019-0627-0

References
1. Ouyang, W. & Zimmer, C. Nat. Rev. Syst. Biol. 4, 105–113 (2017).
2. Falk, T. et al. Nat. Methods 16, 67 (2019).
3. Alipanahi, B., Delong, A., Weirauch, M. T. & Frey, B. J. Nat. Biotechnol. 33, 831–838 (2015).
4. Ouyang, W., Aristov, A., Lelek, M., Hao, X. & Zimmer, C. Nat. Biotechnol. 36, 460–468 (2018).
5. Weigert, M. et al. Nat. Methods 15, 1090–1097 (2018).
6. Moen, E. et al. Nat. Methods https://doi.org/10.1038/s41592-019-0403-1 (2019).
7. Tang, H. et al. BMC Med. Genomics 9, 63 (2016).
8. Esteva, A. et al. Nature 542, 115–118 (2017).
9. Thul, P. J. et al. Science 356, eaal9321 (2017).
10. Lee, H. et al. Nat. Biomed. Eng. 3, 173–182 (2019).

Acknowledgements
This work was funded by the Institut Pasteur. W.O. was a scholar in the Pasteur–Paris University (PPU) International PhD program and was partly funded by a Fondation de la Recherche Médicale (FRM) grant to C.Z. (DEQ 20150331762). W.O. is a postdoctoral researcher supported by the Knut and Alice Wallenberg Foundation (2016.0204) and Erling-Persson Foundation (20180316) grants to E.L. We also acknowledge Investissement d’avenir grant ANR-16-CONV-0005 for funding a GPU farm used for testing ImJoy. We thank the IT department of Institut Pasteur, in particular S. Fournier and T. Menard, for providing access to the kubernetes cluster and DGX-1 server for running and testing the ImJoy plugin engine and for technical support. We thank Q.T. Huynh for maintaining the GPU farm and for advice and assistance during the development of ImJoy. We also thank A. Martinez Casals, P. Thul, H. Xu, A. Aristov, A. Cesnik, C. Gannn, J. Parmar, K.M. Douglass, N. Stuurman, X. Hao, S. Dai, A. Hu, D. Guo, K. Zhou for testing and helping with ImJoy plugin development. We thank E. Rensen for proofreading the manuscript. We thank J. Nunez-Iglesias, S. Mehta, B. Chhun, J. Batson, L. Royer, N. Sofroniew and M. Woringer for useful advice and discussion.

Additional information
Supplementary information is available for this paper at https://doi.org/10.1038/s41592-019-0627-0.