



ImJoy: A computational platform for the deep learning era

Wei Ouyang
Cell Profiling Group, SciLifeLab, KTH, Stockholm
Headed by Emma Lundberg

Overview

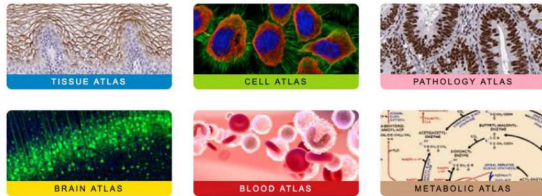
- Human Protein Atlas and HPA image classification competition
- ANNA-PALM: Deep learning for super-resolution microscopy
- ImJoy: A computational platform for the deep learning era

The Human Protein Atlas

THE HUMAN PROTEIN ATLAS

MENU HELP NEWS

SEARCH
e.g. RBM3, insulin, CD38



- Mapping the human proteome using in-house generated proteome-wide collection of antibodies.
- Using large-scale immunostaining and high-resolution microscopy
- Freely available database: www.proteinatlas.org

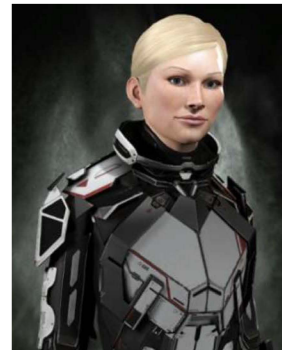
HPA Cell Atlas

Prof. Lundberg

“Foreign Cell Sample” Classification



- >322,000 players on EVE online
- 32,000,000 classifications (70 working years)



Uhlen et al, A tissue based map of the human proteome, Science, 2015
Thul et al, A subcellular map of the human proteome, Science, 2017

Sullivan et al, D, Nature Biotechnol. 2018

Featured Prediction Competition

Human Protein Atlas Image Classification

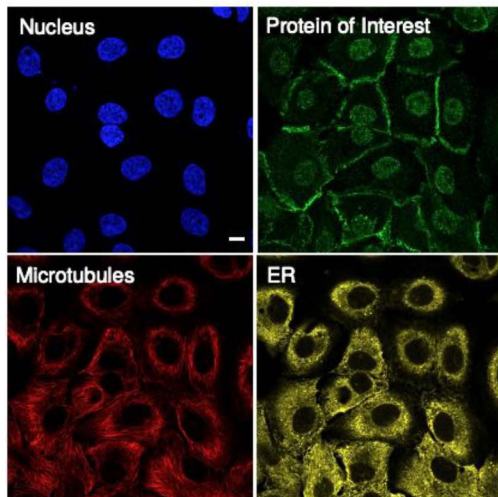
Classify subcellular protein patterns in human cells

\$37,000

Prize Money



Human Protein Atlas · 2,172 teams · 4 months ago



Classifier
→

Multi-label Prediction

Nucleoplasm
Cytosol
Plasma Membrane
Nucleoli
Mitochondria
Golgi Apparatus
Nuclear Bodies
Nuclear Speckles
Nucleoli Fibrillar C.
Centrosome
Cell Junctions
Actin Filaments
...

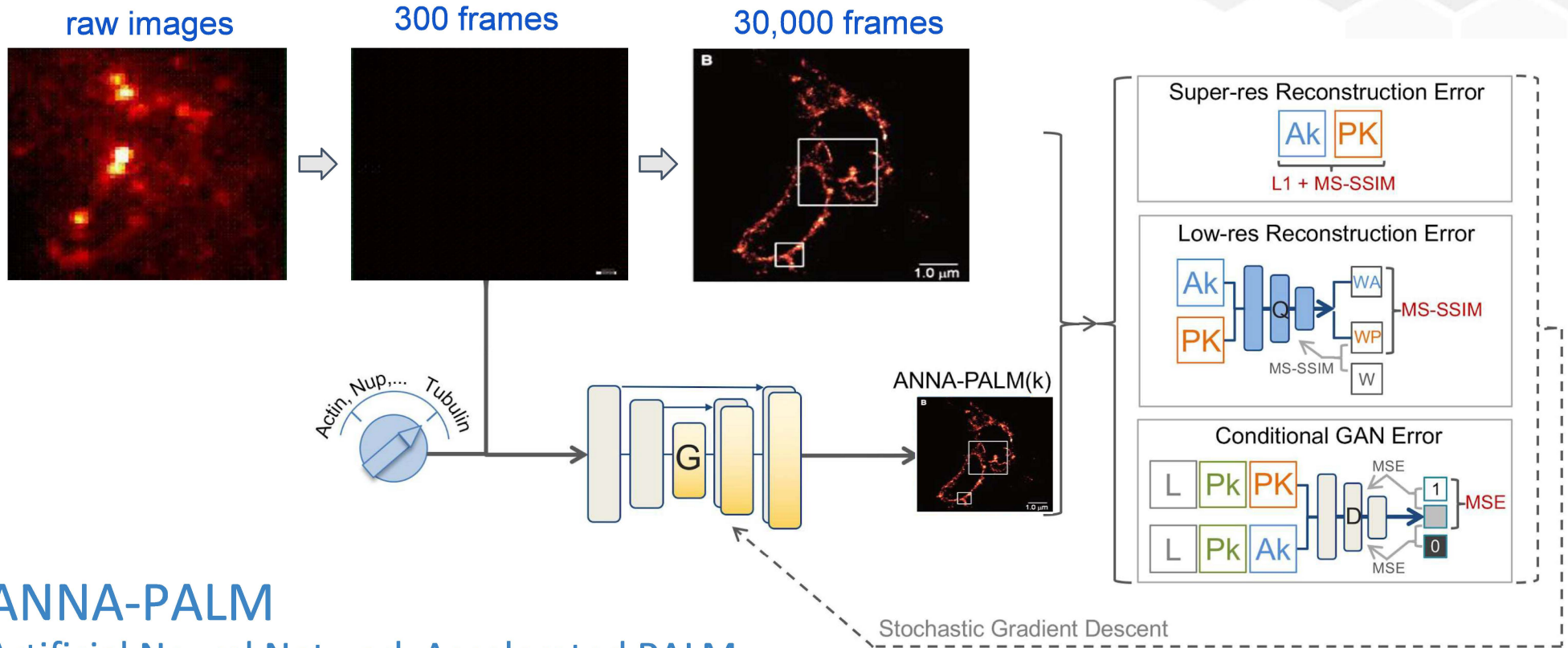
- >120,000 annotated images
- \$37,000 prize money
- 3 months
- >2,000 teams
- >55,000 submissions
- Top model: Densnet 121 + lovaza loss etc.
- 20% higher score than previous record

Deep learning + large image dataset + crowdsourcing!

Ouyang et. al, Nat Methods, 2019

Deep learning accelerated Super-resolution microscopy

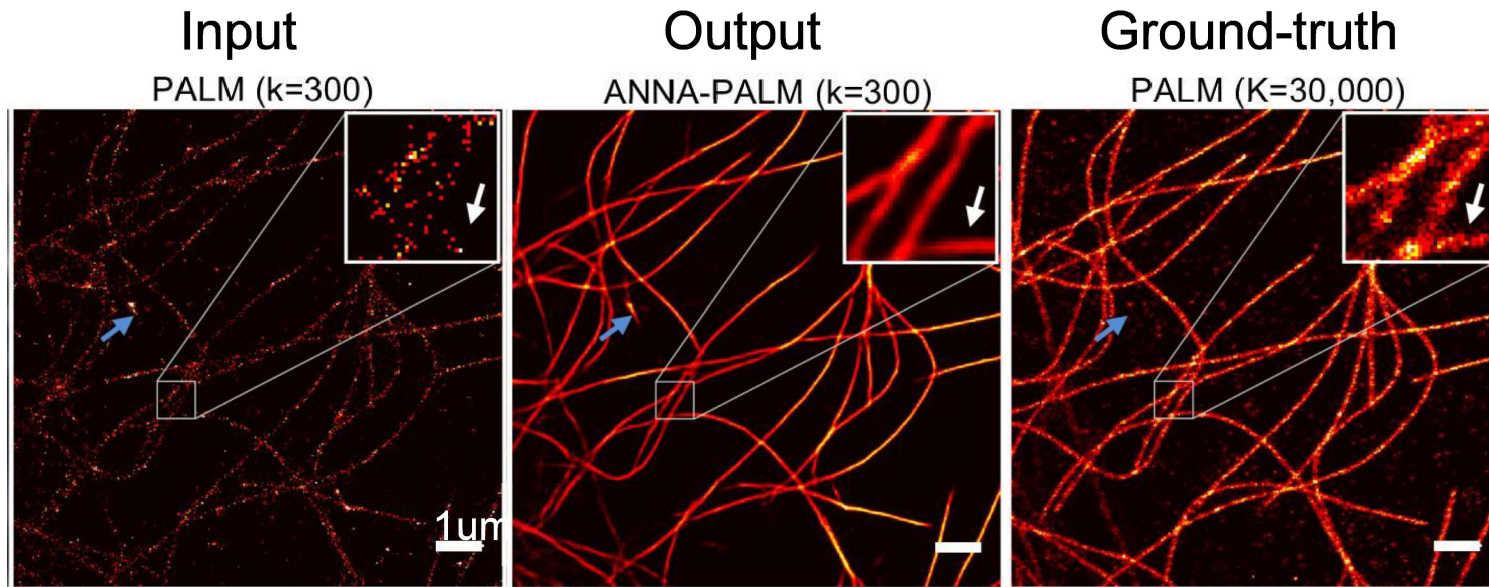
Artificial Neural Network Accelerated PALM/STORM



ANNA-PALM Artificial Neural Network Accelerated PALM

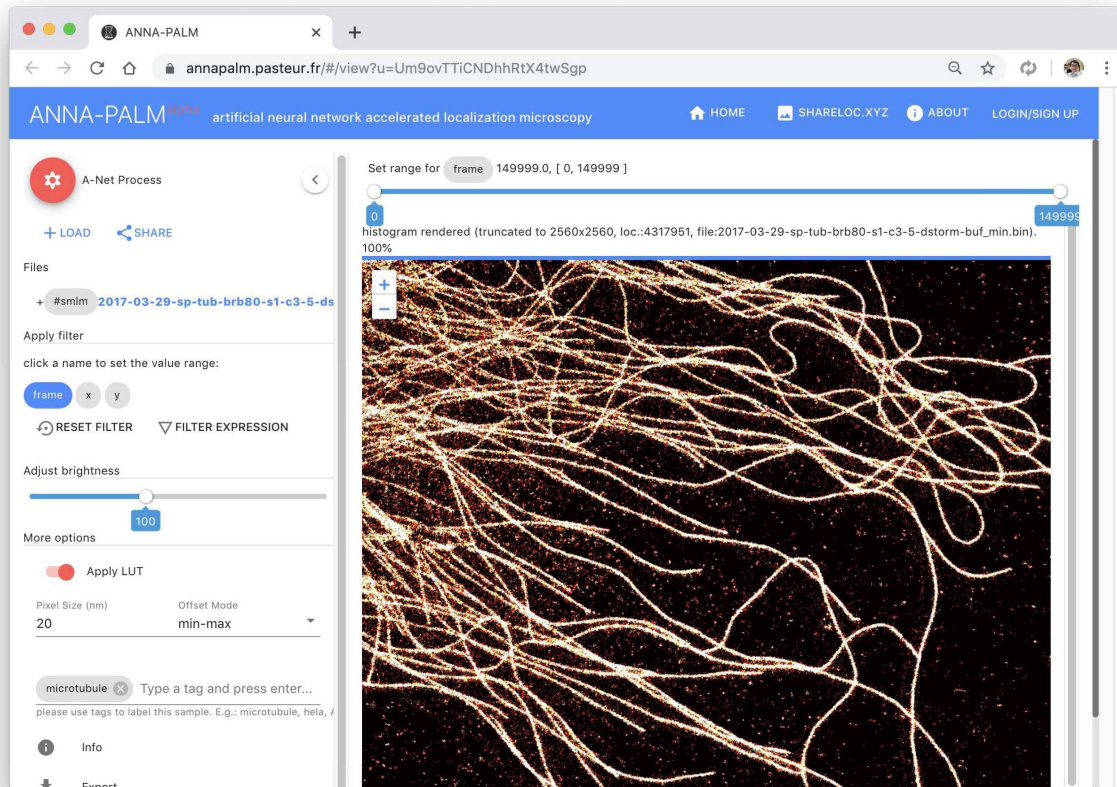
Betzig et al. Science 2006 Ouyang et. al, Nat Biotechnol, 2018

ANNA-PALM: 100x faster SR imaging



Ouyang et. al, Nat Biotechnol, 2018

Web Application for ANNA-PALM



- No installation required
- Not Scalable
- Data transmission
- **Maintenance required!**

- Cost
- Privacy, GDPR

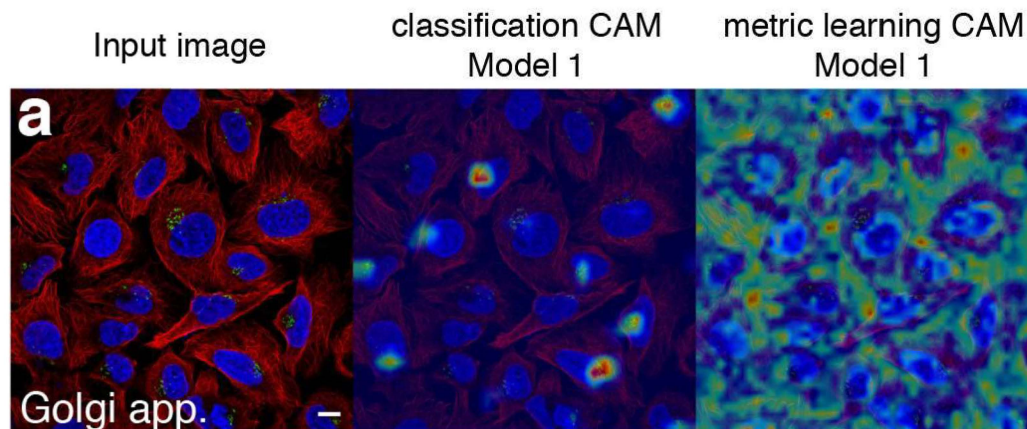
Useful, but requires improvements!

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

Software for the deep learning ara

- Large dataset, remote storage, online database (IDR, HPA)
- Computationally hungry (GPU/TPU/NPU...)
- Interactively inspect, annotate data and train model (human-in-the-loop)

Inspection with Class Activation Maps



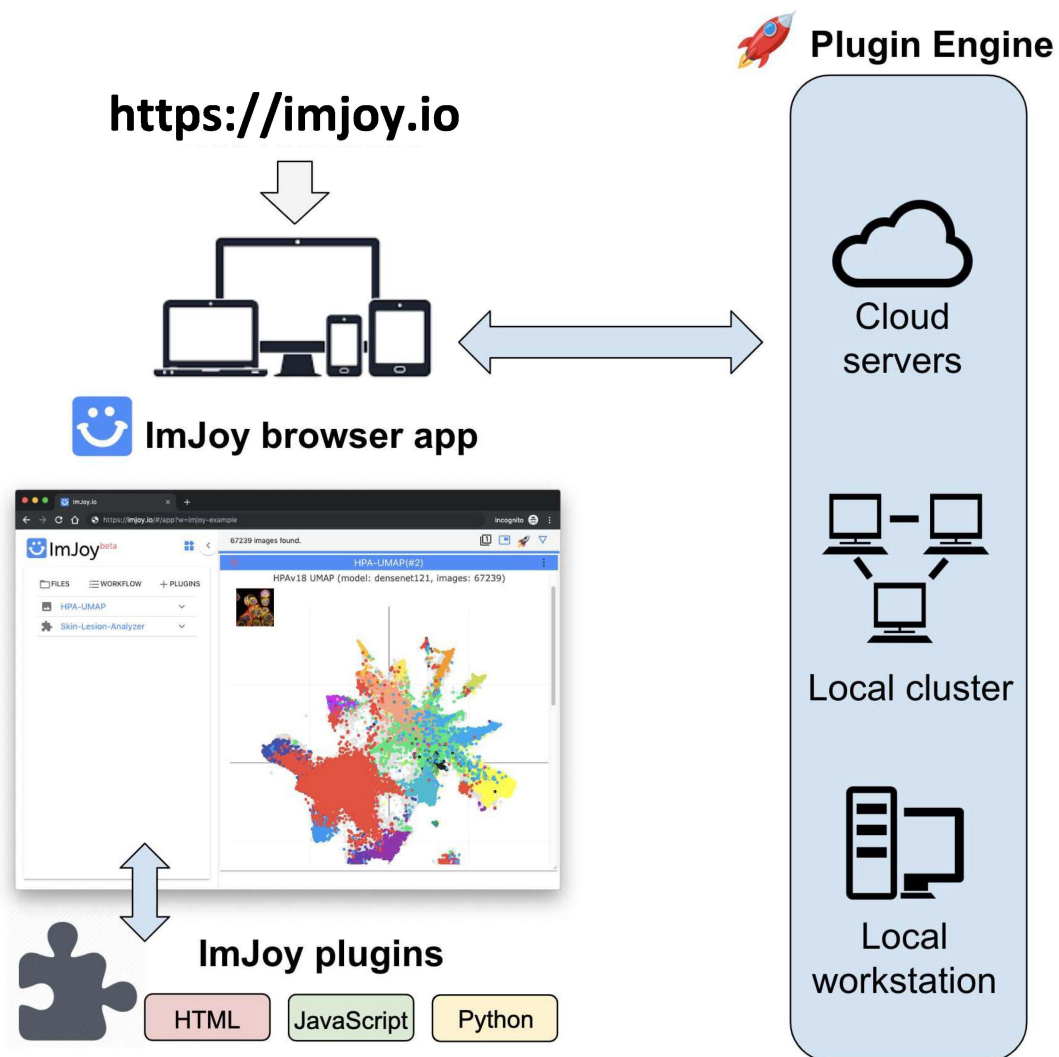
Ouyang et. al, Nat Methods, 2019

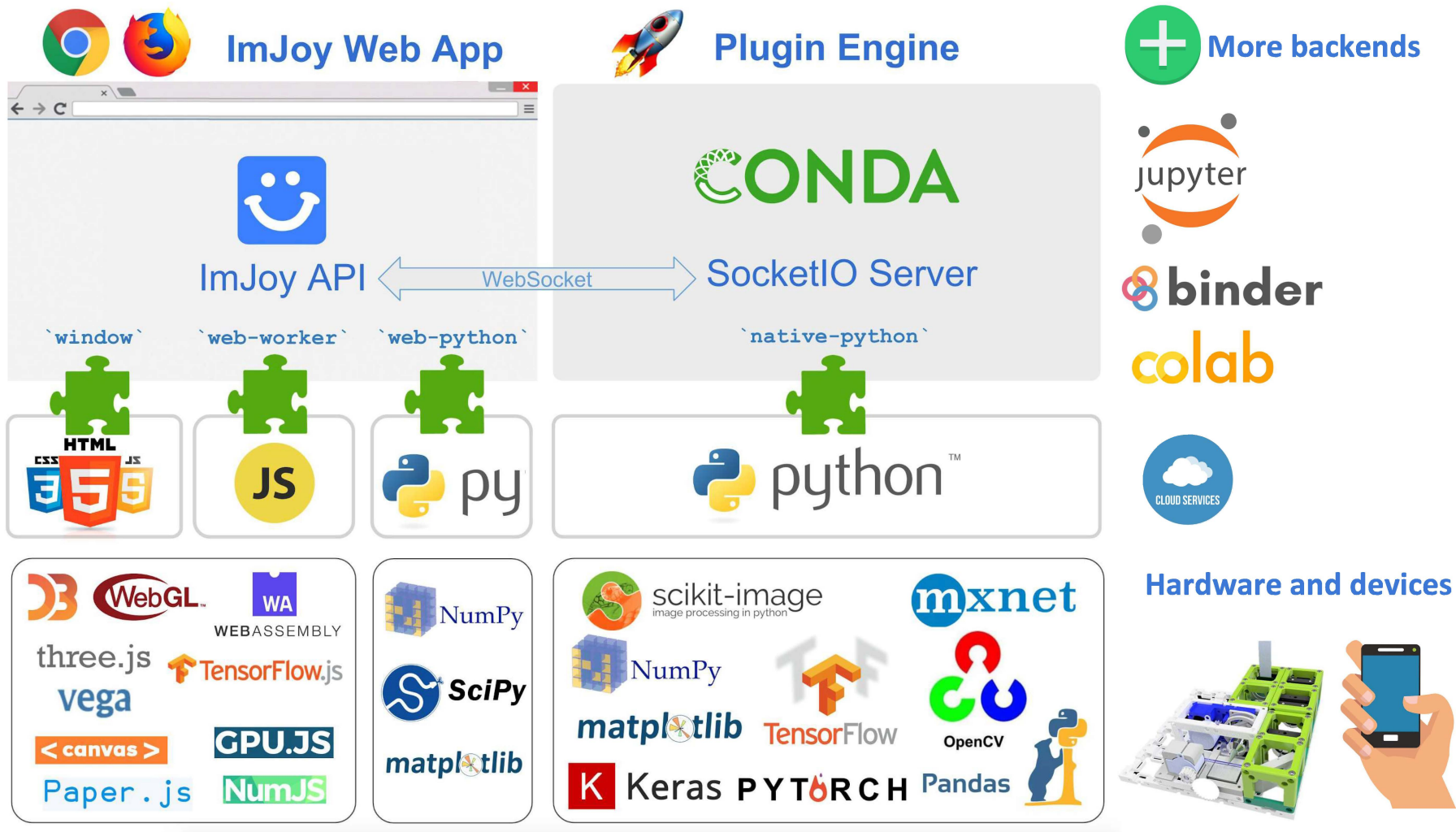
Opportunities

- **Deep learning libraries:** Keras/Tensorflow, Pytorch
- **Mobile** is the biggest platform
 - > 5 billion mobile devices
- **Cloud** computing and remote data access
 - cloud services: storage, GPU, database, serverless, AutoML, Kubernetes
 - Free computation: Google Colab, **MyBinder.org**
 - **Remote Storage: S3, N5, Zarr, Dask Array**
- **Web** standards
 - HTML5 standards, JS libraries, nodejs/npm
 - UI standards (e.g. material design)
 - Progressive Web App, offline support
 - **WebAssembly** (porting C/C++, C#, Rust, Python)
 - **WebGPU** (deep learning in the browser)



- Hybrid computing modes (browser + local + cloud)
- Progressive Web App ⇒ Offline support, mobile support
- Web Assembly/Web GPU ⇒ browser based computing, security





Minimal but powerful!



ImJoy

Key ideas

- Each plugin import and export a set of service functions
- Transparent, symmetrical Remote Procedure Calls (RPC) \Rightarrow across plugins/programming language/host (also see **RPyC** in Python)
- Asynchronous execution \Rightarrow dynamic workflow composition

(Inspired by Crossbar.io, Jailed.js, Tensorflow, Pytorch)



ImJoy

workflow composition: RPC + Async

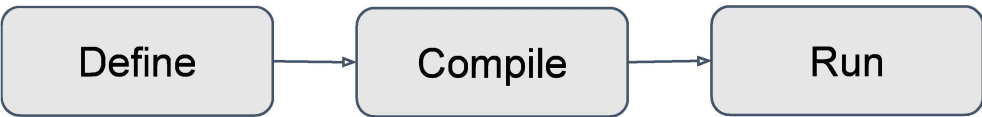
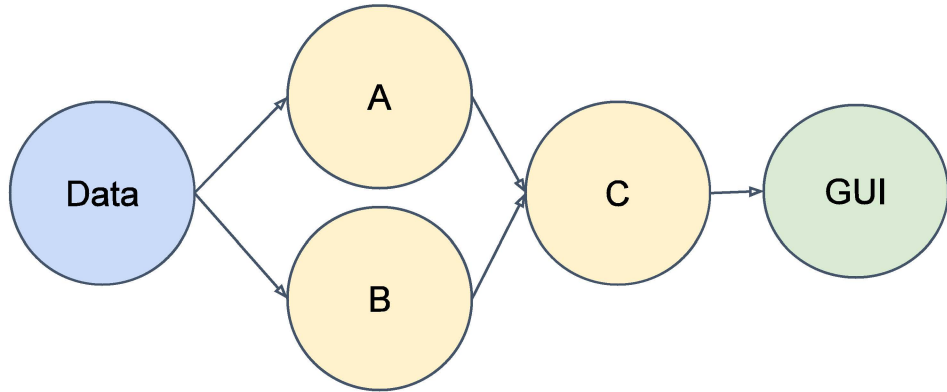
```
1  import asyncio
2  from imjoy import api
3
4  class ImJoyPlugin():
5      async def setup(self):
6          self.pluginA = await api.getPlugin('pasteur.fr/pluginA')
7          self.pluginB = await api.getPlugin('kth.se/pluginB')
8          self.pluginC = await api.getPlugin('pluginC')
```

```
16  async def concurrent_workflow(self, x):
17      promiseA = self.pluginA.process(x)
18      promiseB = self.pluginB.process(x)
19      resultA, resultB = await asyncio.gather(promiseA, promiseB)
20      result = await self.pluginC.process(resultA, resultB)
21      return result
```



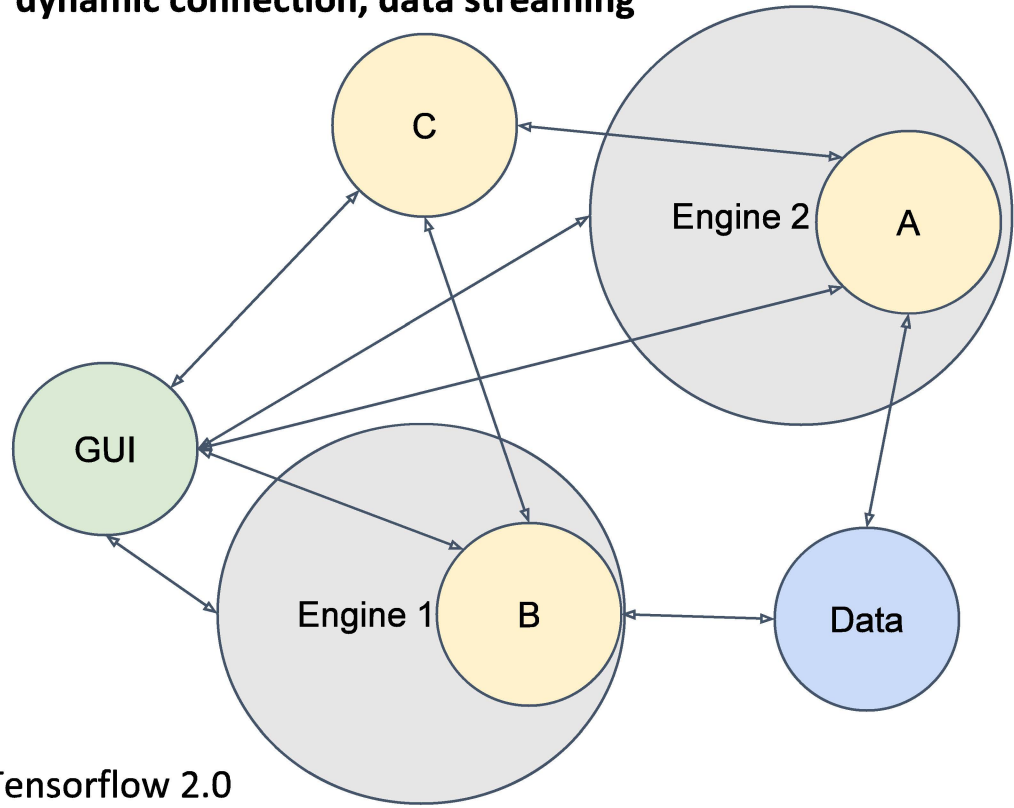
static vs dynamic workflow

Fixed node/connection, file based io
single workflow engine

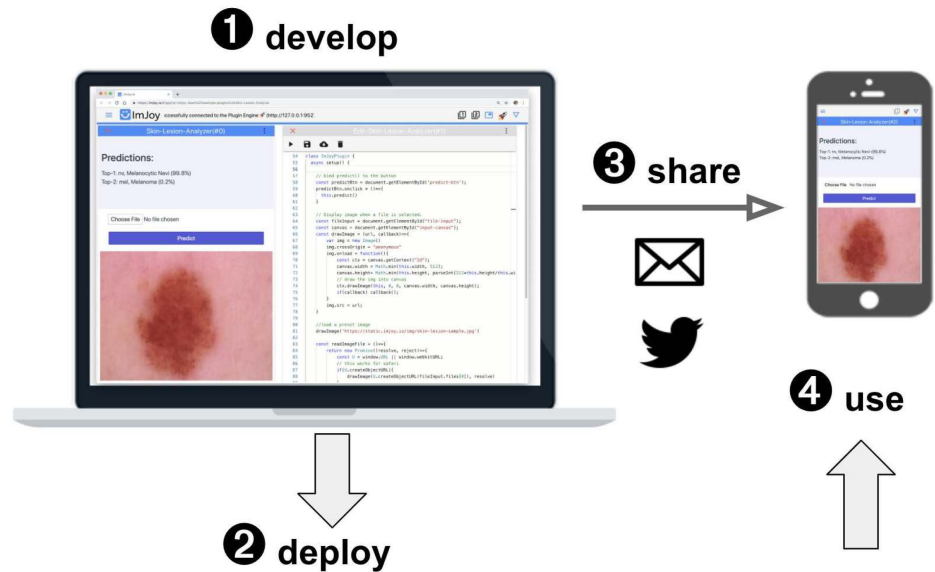


Tensorflow 1.x
Hard to debug!

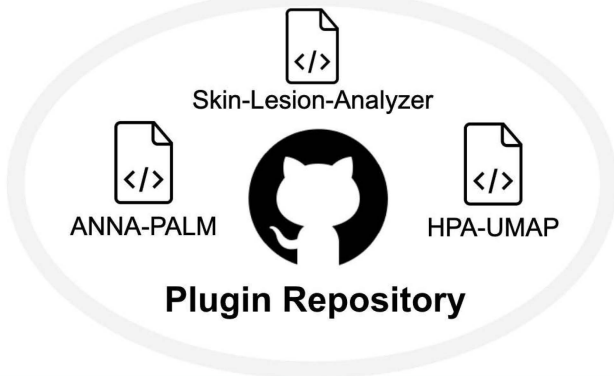
Self-organizing, dynamic node
dynamic connection, data streaming



Tensorflow 2.0
Pytorch



<https://imjoy.io/#/app?p=imjoy-team/example-plugins:Skin-Lesion-Analyzer>



Skin Lesion Analyzer





Noise2self 🚀






Static web app: high scalability & availability, ~ zero cost




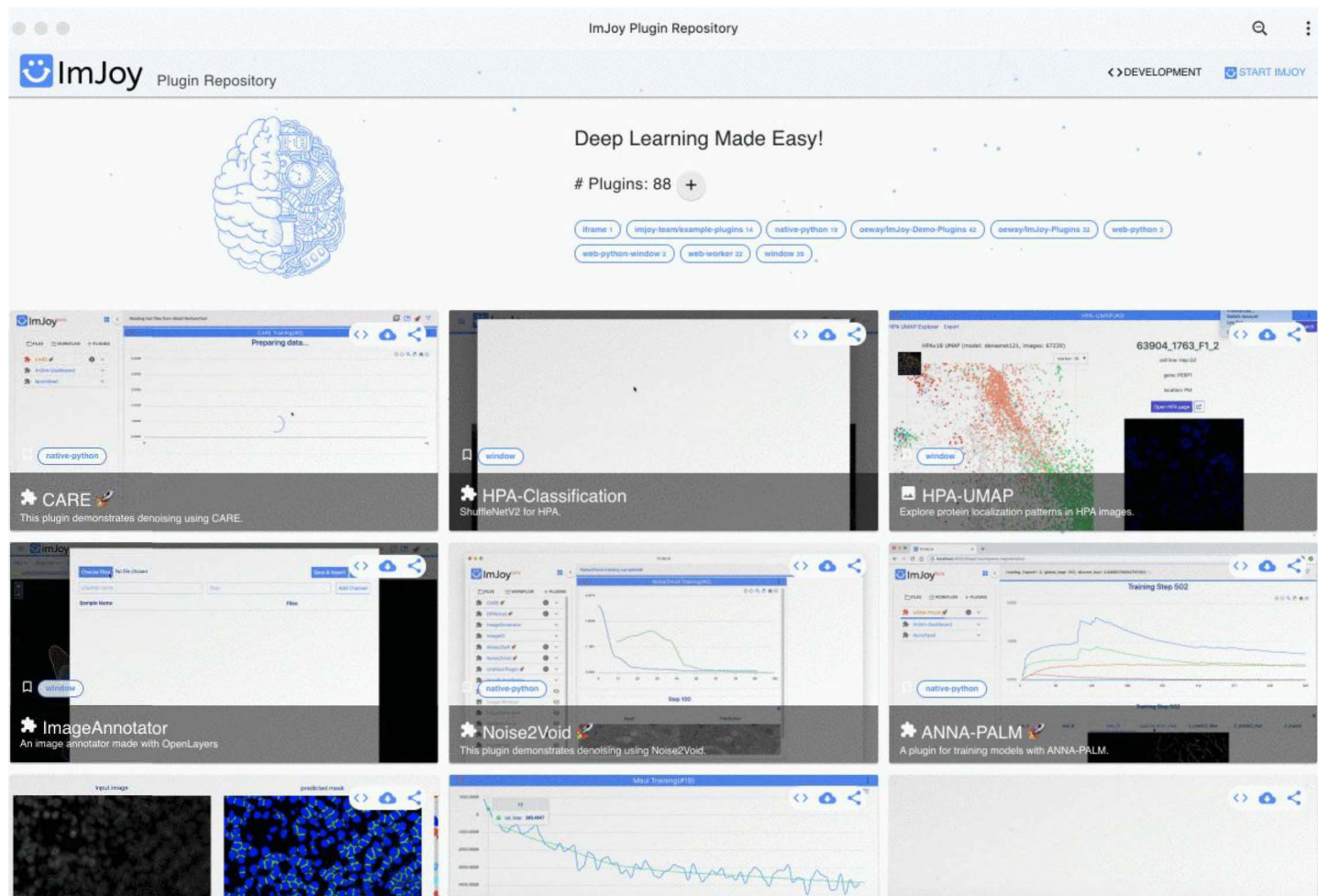
- Plugin repository demo plugins

- [Image Annotator](#)(annotation)
- [HPA-UMAP](#)(visualization)
- [Skin-Lesion-Analyzer](#)(classification)
- [HPA-Classification](#)(classification)

-  [Interactive Plot](#)(basic example)
-  [DeepBindScan](#)(Genomics)

-  [Noise2Self](#)(Denoising)
-  [ANNA-PALM](#)(super-resolution)
-  [CARE](#)(3D image denoising)
-  [DPNUnet](#)(Segmentation)
-  [ImageJ Demo](#)(PyImageJ)

( : requires plugin engine)



Conclusion

- HPA competition: large image database + deep learning
- ANNA-PALM: Deep learning accelerated SR imaging
- ImJoy makes it easier to deploy deep learning models
 - Progressive Web App, Web Assembly, HTML5
 - Extendable Plugin Engines
 - Transparent Remote Procedure Calls
 - Async workflow composition
 - Static, scalable, high availability, almost zero cost

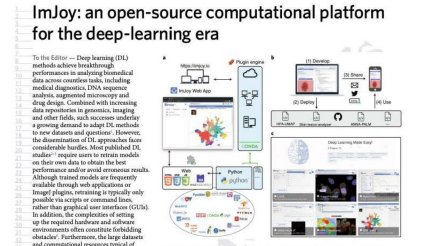
ImJoy is now published in Nature Methods: <https://rdcu.be/bYbGO>
(Deep Learning focused Issue)

Work in progress: data viewers, plugin engine, useful plugins!



FOCUS | CORRESPONDENCE

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



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

To the Editor — Deep learning (DL) methods address longstanding performance in analyzing biological data across multiple fields, including medical diagnosis, DNA sequence analysis, experimental microscopy and drug design. Combined with increasing data repositories in genomics, imaging and other fields, such success underpins 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 install models on their own data to obtain the best performance and/or avoid erroneous results. Although various methods are frequently available through web applications or browser plugins, extending to typically easy-to-use scripts or command lines, rather than graphical user interfaces (GUIs), in addition, the complexity of setting up the required hardware and software environments often constitutes formidable obstacles. Furthermore, the large datasets and computational resources typical of traditional desktop-oriented software that require complex DL and computational GUI and services are partly obscured from scientists, but raise privacy and confidentiality issues that can be prohibitive for smaller labs.

Meanwhile, deploying scientific software to mobile platforms can make them accessible to billions of people, enabling large-scale biomedical research and clinical services. These opportunities and challenges call for an open computational framework.

Recognizing this need, we developed ImJoy (ImJoy.com), an open-source platform designed to deliver advanced, yet easy-to-use data analysis tools, especially based on DL (Fig. 1). Supplementary Note 1 in ImJoy's user manual, available at <https://www.imjoy.io>, provides a first and stable user experience across all major platforms, including desktop 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. 1). 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 connections: once a plugin has 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 available to process sensitive data. ImJoy makes it easy to build applications using existing software libraries, integrate with unlabelled platforms, such as Jupyter and Ingestor (Fig. 2), and provide diverse various computing infrastructures (Fig. 1). Supplementary Note 2 in ImJoy provides a built-in code editor for developing and testing plugins. A plugin asset consists of a single file that can be hosted on GitHub, and installed with a single, easily shareable hyperlink (Fig. 1). Plugin operations can be made easily accessible by gallery view (Fig. 1). Next, we highlight ImJoy's main capabilities, along with example plugins (Supplementary Note 3). For detailed documentation, including materials on porting existing tools to ImJoy, see <https://www.imjoy.io>.

By developing scientific applications, such as those in ImJoy, ImJoy allows users to build full and interactive applications (Fig. 1). For example, the ImageJ-extension plugin allows visualization of images, a pre-processor for training segmentation methods and can run on touch-screen devices (for example, smartphones). The IOP (Image Processing) visualization plugin localizes features computed by the IOP (Image Processing) plugin. Such interactive visualizations are implemented by requiring massive databases and interacting with cloud services.

Acknowledgements



- Thank all the Kaggle participants for their contributions.



Lundberg Lab

Ulrika Axelsson
Anna Bäckström
Jenny Fall
Christian Gnann
Martin Hjemare
Diana Mahdessian
Anna Martinez Casals
Rutger Schutten
Charlotte Stadler
Devin Sullivan
Peter Thul
Casper Winsnes
Hao Xu
Lovisa Åkesson
Anthony Cesnik

Human Protein Atlas

- Mathias Uhlén
- Fredrik Pontén
- Adil Mardinoglu
- Kalle von Feilitzen
- Jan Mulder
- Cecilia Lindskog
- Burcu Ayoglu

Collaborators

- Manu Leonetti, CZ Biohub
- Kathryn Lilley, Cambridge Univ
- Oana Carja, CMU
- Attila Szantner, MMOS
- Andie Nordgren, CCP games
- Bergur Finnbogasson, CCP games
- EVE Online players



SciLifeLab



Vetenskapsrådet



Acknowledgements

 @weioyang

Imaging & Modeling Unit @ Institut Pasteur

- Christophe Zimmer
Andrey Aristov, Benoît Lelandais, Christian Weber, Elena Rensen,
Florian Muller, Jyotsana Parmar
Mickael Lelek, Maxime Woringer, Xian Hao)
- Cécile Leduc, Sandrine Etienne-Manneville
- Sandrine Lévêque-Fort, Nicolas Bourg
- Orestis Faklaris, Julien Sellès



The IT service of Pasteur, Inception GPU lab

(**Stéphane Fournier**, **Thomas Menard**, Jean-Baptiste Denis, Tru Huynh, et. al.)

Thank you!

Useful links for ImJoy



Source code and documentations

- [documentation](#), [API functions](#)
- [main repo](#), [plugin-engine](#)

Example Plugins

- [official plugin repository](#), [example plugins](#)

Getting Help

- [Image.sc Forum](#)
- [ImJoy Slack](#)

Bug Report

- [ImJoy Issues](#)

Preprint on Nature Methods: <https://rdcu.be/bYbGO>