



ILLUSTRATION BY THE PROJECT TWINS

# NAPARI GIVES PYTHON ITS OWN IMAGE VIEWER

The image viewing and analysis software napari has filled a gap in the programming language’s scientific ecosystem. **By Jeffrey M. Perkel**

**J**osh Dorrington has become adept at viewing the jet streams. He plots fast-moving rivers of air at different atmospheric altitudes and positions the charts side by side. “You get pretty good at looking at all these cross-sections and working out what it implies,” says Dorrington, an atmospheric physicist at the University of Oxford, UK. But compared with computerized visualizations, this ‘manual’ method is slower and “it’s not as interactive”.

So when it came to visualizing atmospheric blocking events, in which stationary air masses of high pressure deflect the jet stream and lead to extreme weather events, Dorrington tried an alternative approach. Using the image viewer napari, he animated the streams in 3D as they

flowed over the Northern Hemisphere – usually, but not always, in sync with each other. “One afternoon spent hacking stuff together and I already have a short visualisation of the lower and upper level Atlantic jet streams!” he tweeted in August.

napari is a free, open-source and extensible image viewer for arbitrarily complex (*n*-dimensional) data that is tightly coupled to the Python programming language (see [napari.org](https://napari.org)). It is the brainchild of microscopist Loïc Royer, at the Chan Zuckerberg Biohub in San Francisco, California; Juan Nunez-Iglesias, who develops image analysis at Monash University in Melbourne, Australia; and Nicholas Sofroniew, who leads the imaging-technology team at the Chan Zuckerberg Initiative (CZI)

in Redwood City, California.

The team created the software in 2018 to fill a gap in the Python scientific ecosystem. Although Python is the lingua franca of scientific computing, it had no visualization tools that could handle *n*-dimensional data sets. As a result, data analysis usually required a tedious back and forth between Python and other tools (such as the Java-based image-analysis package ImageJ) as researchers alternately manipulated their data and then visualized them to see what had happened.

napari – the name refers to a Pacific island village midway between the developers’ bases in San Francisco and Melbourne – features a simple graphical interface with a built-in Python console in which images can

be rendered, rotated and manipulated in 2D or 3D, with additional dimensions, such as the succession of temporal 'slices' in a time series, accessible using sliders beneath the image window. If available, graphics-processing units can be used to accelerate the software. "We make sure that we actually use the computer to its full capacity," Royer explains. (Image) users can also work in Python using PyImageJ; see [pypi.org/project/pyimagej](https://pypi.org/project/pyimagej)).

Adobe Photoshop-like layers allow users to overlay points, vectors, tracks, surfaces, polygons, annotations or other images. A researcher could, for instance, open an image of a tissue in napari, identify cell nuclei with a click of the mouse, retrieve those points in Python and use them to 'seed' a cell-segmentation algorithm, which identifies cell boundaries. By then pushing the results to napari as a new layer on the original image, they can assess how well the segmentation process worked.

Xavier Casas Moreno, an applied physicist at the Royal Institute of Technology in Stockholm, has embedded napari in an instrument-control system, called ImSwitch, that he built to control his lab's super-resolution microscopes. Using layers, researchers can overlay images taken from different angles by multiple cameras and sensors. "That's a challenge that we couldn't solve before," Moreno says. A custom napari control panel, or 'widget', built into ImSwitch, allows users to move one image relative to the other to align them.

Users can also extend napari using plug-ins: more than 100 are available on the napari hub (see [napari-hub.org](https://napari-hub.org)). These include plug-ins for reading and writing files, microscope control and cell segmentation. Carsen Stringer, a computational neuroscientist at the Howard Hughes Medical Institute's Janelia Research Campus near Ashburn, Virginia, created a cell-segmentation plug-in called CellPose, which makes the power of Python deep-learning algorithms available through a simple graphical interface. Stringer and her colleagues use CellPose to measure gene expression and calcium signalling cell by cell in neural tissue.

Robert Haase, who develops biological image analysis at Dresden University of Technology in Germany, has built several plug-ins for napari. These include one for his py-clEsperanto assistant, which aims to accelerate image processing and remove programming-language barriers between different image-viewing systems, and one called natari – a portmanteau of napari and Atari (the videogame developer) – in which users play a game that involves removing cells from a field of view by shooting them.

Nunez-Iglesias has even written a plug-in that allows users to control napari through a music instrument digital interface (MIDI) board, which typically would be used to tailor digital audio output. In a video demonstration,

Nunez-Iglesias combines a MIDI controller and Apple iPad to manually segment cells by drawing on the image on the tablet with an Apple Pen, and moving from layer to layer in napari by turning a dial on the MIDI controller. "I just like hardware controllers and physical feedback more than virtual user interfaces," he explains.

At version 0.4.12, napari is still in development, with support from the CZI science imaging programme, which hosts the napari hub and provides several staff members to the project, and code contributed by many developers in the scientific community. CZI has also earmarked US\$1 million to support the napari Plugin Accelerator grant programme: 41 awards were announced last month.

Key development goals include improving installation, documentation, the plug-in interface and the handling of large 3D data sets. Some experiments, such as neural connectomics studies, produce data sets so massive

that the software slows to a crawl, says Nunez-Iglesias. A 'lazy-loading' scheme, in which data are loaded on demand, should alleviate that bottleneck, he says.

Nunez-Iglesias estimates that napari's user base is "in the thousands". And it's growing fast, even among non-microscopists. Users have demonstrated applications in geophysics and structural biology, for instance. And DeepLabCut, a popular tool for capturing the poses of animals as they move, has announced plans to replace its viewer with napari.

Hung-Yi Wu, a systems biologist then at Harvard Medical School in Boston, Massachusetts, summed up the excitement the project is generating in a tweet in 2020. "@napari\_imaging is awesome," he wrote: "I felt like [I was] using the first iPhone".

**Jeffrey M. Perkel** is technology editor for *Nature*.

# 'ELECTRONIC MUD' KEEPS SUPERB TIME

Reprogrammable arrays of logic elements shine when speed matters. **By Jeffrey M. Perkel**

**M**ichael Reiser is, as he puts it, "fanatical about timing". A neuroscientist at the Howard Hughes Medical Institute's Janelia Research Campus in Ashburn, Virginia, Reiser studies fly vision. Some of his experiments involve placing flies in an immersive virtual-reality arena and seamlessly redrawing the scene while tracking how the insects respond. Modern PCs, with their complex operating systems and multitasking central processing units (CPUs), cannot guarantee the temporal precision required. So Reiser, together with engineers at Sciotex, a technology firm in Newtown Square, Pennsylvania, found a piece of computing hardware that could: an FPGA.

An FPGA, or field-programmable gate array, is basically "electronic mud", says Bruno Levy, a computer scientist and director of the Inria Nancy Grand-Est research centre in Villers-lès-Nancy, France. It is a collection of hundreds or even millions of unconfigured logic elements on a silicon chip that, like clay, can be 'moulded' – and even re-moulded – to accelerate applications ranging from genomic alignment to image processing to deep learning.

Suppose that a researcher needs to quickly process data streaming off a camera in chunks of 1,000 bits. Most modern CPUs have 64-bit

processors and would have to break the problem into smaller pieces. But it's possible to configure an FPGA to do that calculation in a single step, says Inria Nancy computer scientist Sylvain Lefebvre. Even if each FPGA step is slower than its CPU counterpart, "it's actually a win, you're going faster", he says, because the problem isn't broken down. FPGAs excel at applications requiring precise timing, speed-critical algorithms or low energy consumption, he adds.

**"You describe how the data moves between registers each clock cycle, which is not how most developers think."**

Javier Serrano, manager of electronics design and low-level software at CERN, Europe's particle-physics laboratory near Geneva, Switzerland, and his colleagues used FPGAs, plus White Rabbit – a bespoke extension to the Ethernet networking protocol – to create a system that can capture instabilities in the Large Hadron Collider particle beam with nanosecond precision.

At Queens University Belfast, UK, computer