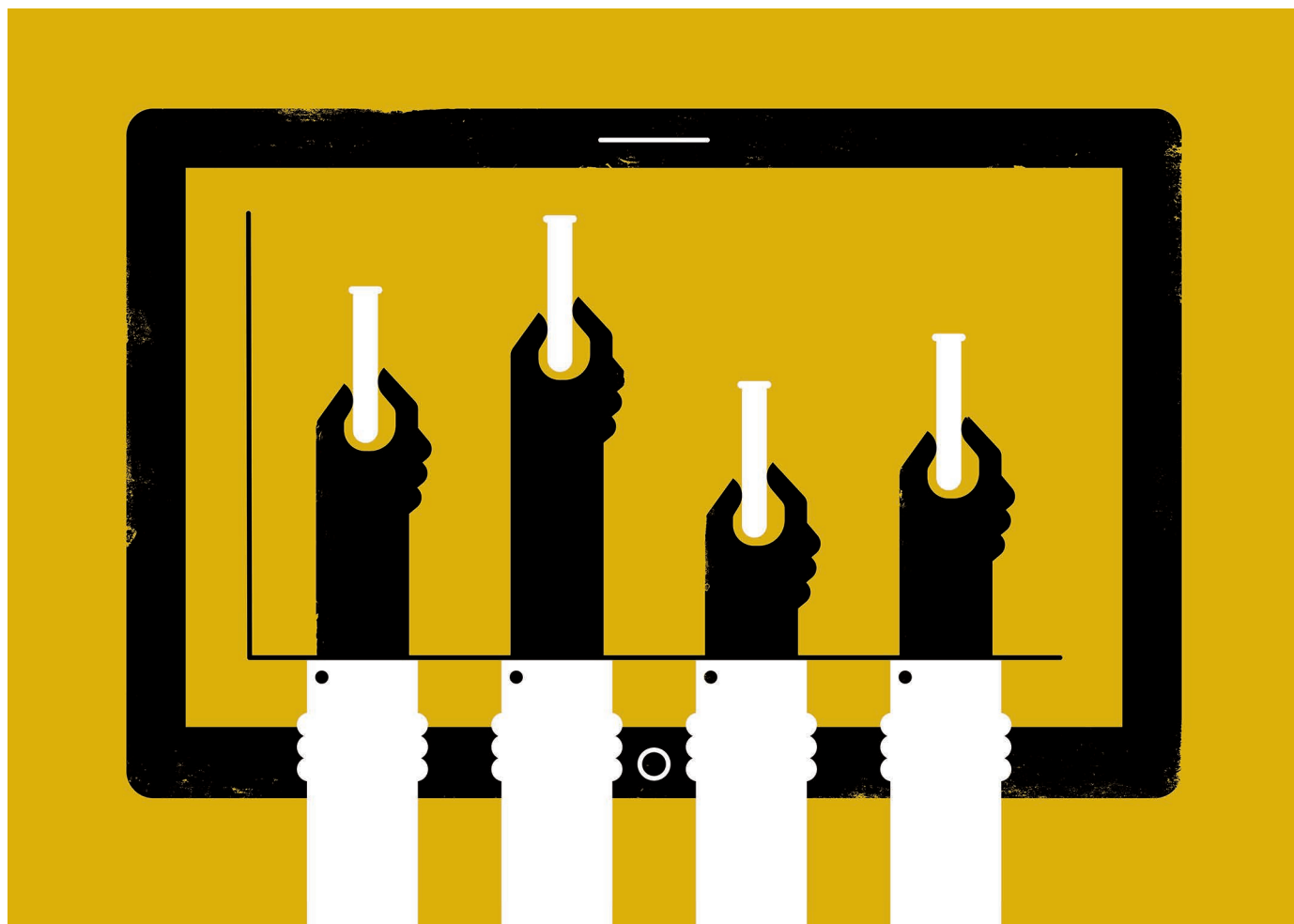


THE FUTURE OF SCIENTIFIC FIGURES

New tools for building interactive figures and software make scientific data more accessible, and reproducible.

ILLUSTRATION BY THE PROJECT TWINS



BY JEFFREY M. PERKEL

As Benjamin Delory started his paper documenting a new analysis pipeline to quantify plant morphology, he realized that one of the figures could pose a problem.

The paper proposes a framework to compute ‘persistence barcodes’ that describe the branching structure of plant root systems¹. The challenge was how to illustrate it.

The barcode’s underlying algorithm “is

continuous and dynamic”, says Delory, a post-doctoral researcher at Leuphana University of Lüneburg in Germany. “And the best solution to show something dynamic is to animate it”

Scientific figures are typically rendered as static images. But these are divorced from the underlying data, which prevents readers from exploring them in more detail by, for instance, zooming in on features of interest. For genomics needing to cram millions of data points into dense visuals a few centimetres big, this can be particularly problematic.

The same is true for researchers working with computational algorithms. Scientists often post software on open-source repositories such as GitHub, but getting the code to run properly is easier said than done. Reviewers and other interested parties often require extra software and configuration to make the algorithms work.

Some journals now bridge that gap by supporting interactive figures and code. One of those is *F1000Research*, which last year partnered with the computing firm Plotly in Montreal, Canada, and the Code Ocean platform ►

► in New York City. These capabilities, as well as *F1000Research*'s open-access ethos, led Delory and his collaborators to submit their paper there. It was published in January¹.

THE INTERACTIVE PUBLICATION

Interactive graphics that allow readers to delve into a story's underlying data are frequent features on websites such as those of the *New York Times* and *fivethirtyeight.com*, but are less common in scientific publishing.

F1000Research's 'living figures' — interactive charts introduced in 2014 that could be continually updated with new data — were laborious to produce and unscaleable, says senior publishing editor Thomas Ingraham. Plotly lets users build and share visualizations ranging from scatter plots and line graphs to contour plots and maps. The resulting images allow users to zoom in on data, pan across images and mouse-over points to see the plotted values. Student subscriptions start at US\$59 per year. Open-source libraries allow researchers to create free Plotly graphics from R, MATLAB, Python and Julia code.

Code Ocean is free for academics for 10 hours of computation time per month and 50 gigabytes of storage; paid tiers start at \$19 per month. It brings together code, data, results and the computing environment used to execute them in a self-contained 'compute capsule' that replicates the author's computational configuration. Other users can download, modify and run that code either from *codeocean.com*, or through a widget in the paper.

F1000Research has now published six papers with live Plotly graphs and five with a Code Ocean widget. And this year, it plans to add support for interactive protein–protein interaction maps, which are produced using the network-mapping tool Cytoscape.

Researchers need not be put off by the perceived complexity. According to computational biologist Xijin Ge at South Dakota State University in Brookings, who has included interactive Plotly graphs in one of his papers², creating those figures requires just one extra line of code per figure. Tom DeCarlo, a coral researcher at the Oceans Institute and School of Earth Sciences at the University of Western Australia in Crawley, has created six Code Ocean projects for journals including *Paleoceanography and Paleoclimatology* and *Biogeosciences*. "I thought it was really important for scientific communication and reproducibility," he says.

OPEN-SOURCE SOLUTIONS

For those seeking open-source computational alternatives, a tool known as Binder can convert any public GitHub repository containing a Jupyter notebook (documents that interleave text, code and data) or R code into a package that users can run from their browser. Users simply type the notebook repository address into the search bar at *mybinder.org*, and the program creates a shareable interactive workspace.

"It really lends itself to reproducibility and ease of use," says Carol Willing, a Binder project team member at California Polytechnic State University (Cal Poly) in San Luis Obispo.

Such tools also simplify peer review, says Tim Head, a member of the Binder project team in Zürich, Switzerland. Head was frustrated that he couldn't make the software work when asked to review a journal article. "Had they sent me a Binder link, we'd be done by now," he says.

Open-source options also exist for creating interactive images, including Bokeh, HTML widgets, pygal and ipywidgets. Most are used programmatically, generally within either R or Python code, which is commonly used in science. Coders can, for example, use ipywidgets to drop interactive 3D plots, maps and molecular visualizations into Jupyter notebooks. Another option, which is written in JavaScript, is Vega-Lite. Because that language is less popular in science, Brian Granger at Cal Poly and Jake VanderPlas at the University of Washington in Seattle developed a Python interface called Altair to make it more accessible.

Whereas most of these tools tend to provide functions for specific graph types, Vega-Lite and Altair are flexible 'grammars' that describe, for instance, how variables map to different visual features, such as colour or shape. They also allow graphs to be linked, such that when users select a region of one plot, the displays of its neighbours update accordingly. "It lets us actually explore relationships in a multidimensional way," says Jeffrey Heer, a computer scientist at the University of Washington whose lab developed Vega-Lite.

Two other products let researchers create interactive apps that make use of widgets such as drop-down menus and slider controls to blend data, graphics and code: Shiny, made by RStudio in Boston, Massachusetts, for R, and Plotly's Dash for Python. They work by transmitting the user's widget actions to a remote server, which runs the underlying code and updates the page.

The resulting apps can make data and tools accessible to researchers who are uncomfortable with programming. For instance, graduate student Tal Galili worked with colleagues at Tel Aviv University to develop a Plotly-based toolbox to build interactive heat maps from uploaded data sets, as well as a Shiny interface that runs the code behind the scenes. Mine Çetinkaya-Rundel, a statistician at Duke University in Durham, North Carolina, has built Shiny resources for her undergraduate statistics courses to help her to illustrate difficult concepts during lectures.

"It's nice to just pull that up and say, 'okay, now that we've introduced this thing, what happens when we move around the widgets?'" she says.

Publishing such integrations on journal web pages involves making changes to authoring

tools, editorial workflows and infrastructure. It might also involve entrusting scientific data to third parties, who cannot always guarantee their permanence.

To help address this, open-access publisher *eLife*'s Reproducible Document Stack project aims to create an end-to-end tool set for authoring, submitting and publishing documents that are computationally reproducible, says Giuliano Maciocci, who leads product development at *eLife*. The plan is to encapsulate many of a paper's core scientific 'artefacts' — its text, figures, code, data and computational environment — in a single downloadable object, he says. To encourage adoption, the journal is making the stack open source.

MAKING HEADWAY

Several other journals and publishers now support Code Ocean integration, including *GigaScience*, IEEE, SPIE, Cambridge University Press and Taylor & Francis. The *Journal of Cell Biology*'s JCB DataViewer, based on open-source OMERO software, lets readers explore raw microscopy images rather than the processed, compressed files they typically see. A related tool, the Image Data Resource, offers similar functionality for papers published in any journal. *Nature*, too, has published interactive figures, for instance in a paper describing the Encyclopedia of DNA Elements project³. A spokesperson says that the journal is investigating several other options for interactive code and figures. In the meantime, researchers often link to external visualizations from their articles.

As more journals embrace interactivity, the online presentation of scientific information could fundamentally change, representing a win for reproducibility, says Erez Lieberman Aiden of the Baylor College of Medicine in Houston, Texas, who published interactive chromatin interaction maps in a recent *Cell* paper⁴. Static figures are just one perspective on the data. "Informed readers need the ability to draw their own conclusions," he says. "The act of reading a paper in 1974 and the act of reading a paper in 2017 shouldn't be the same act." ■

Jeffrey M. Perkel is *Nature*'s technology editor.

1. Jung, D. & Ge, X. *F1000Research* **6**, 1969 (2017).
2. The ENCODE Project Consortium. *Nature* **489**, 57–74 (2012).
3. Rao, S. S. P. et al. *Cell* **171**, 305–320 (2017).

CORRECTION

The Toolbox 'A bioinformatics workshop in a box' (*Nature* **552**, 137–138; 2017) erroneously affiliated Robert Gentleman with the Cambridge campus of Harvard University. He was, in fact, at the Harvard T. H. Chan School of Public Health in Boston, Massachusetts.

CORRECTION

The Toolbox article ‘The future of scientific figures’ (*Nature* **554**, 133–135; 2018) implied that Benjamin Delory developed the persistence barcode method. In fact, he and his colleagues developed an analysis pipeline that relied on and adapted an existing method.