

Closing the divide between analysis and publication: The notebook pub

We're experimenting with treating our computational notebooks as publications themselves. This approach reduces publication burden, encourages faster publishing, and builds in reproducibility. Scientists can publish with minimal extra effort.

Published Mar 9, 2025



Arcadia Science

DOI: 10.57844/arcadia-ca21-23bb

Purpose

Much of the research work we do at Arcadia is computational. Our scientists often develop their core ideas in Jupyter Notebooks, a popular tool that's great for rapid exploration and internal sharing. They provide a one-stop-shop for writing code, visualizing results, and documenting our thinking. But we've noticed that when the work is ready to be shared, there's still a barrier to converting these computational products into pubs, adding unneeded friction between how we conduct computational research and how we share it with the community.

This disconnect perfectly illustrates why we recently shifted to a more scientist-driven publishing model at Arcadia [1][2]. Rather than having our publishing processes dictate how scientists need to package their work, we're empowering them to share in ways that feel most natural and useful. Continuing this experiment with publishing, we wondered: what if we could directly share our notebooks, preserving the natural flow of research while making the work immediately useful to others? After following this line of inquiry, we're introducing a new publishing format at Arcadia: the **notebook pub**.

Notebook pubs treat the scientist's working notebook as the publication itself. Rather than maintaining separate documents for analysis and publication, the

notebook serves as a single source of truth where code, results, and narrative coexist. When ready to share, scientists transform their notebook into a publication with minimal additional effort, focusing on its accessibility and reusability.

We've developed a template that works for Arcadia pubs, and we encourage you to adapt it to suit your needs.

- This pub is part of the **model creation effort**, "Reimagining scientific publishing." Visit the model narrative for more background and context.
- Check out our **first notebook pub**, "Paired residue prediction dependencies in ESM2."
- Experiment with our **notebook pub template** by cloning this GitHub repo.

Background

Research is becoming increasingly computational, but there remains a persistent gap between the computational tools scientists use for analysis and the publication formats used for sharing work. To bridge this gap between analysis and publication, we've developed a pub format for Arcadia that we call the "notebook pub." We've developed a workflow that automatically converts our Jupyter Notebooks into hosted publishable documents. The resulting pub is a webpage that preserves all the interactive elements of the original notebook while adding necessary publishing features like licensing information and commenting capabilities.

This initiative aligns with a broader community-wide movement toward "executable papers." Several emerging publishing platforms (e.g., NeuroLibre, Nextjournal, Notebooks Now!, and Physiome) now support direct notebook-to-publication conversion as a result of various notebook conversion tools (e.g., Jupyter Book, Quarto, and Curvenote). In the same vein, we've created a lightweight notebook publishing format specifically tailored for Arcadia publications.

In this pub, we outline the benefits of this strategy, how we've approached it from a technical perspective, what sort of feedback we'd like, and what we're trying next.

Notebook pubs accelerate our research and the community's science

When we close the gap between how science is done and how it's shared, there should be two clear benefits to the research ecosystem — full reproducibility and earlier information-sharing.

CHALLENGE 1: Scientific publications should provide a clear, reproducible path from the first byte of raw data to the last period of the final sentence.

At its core, computational analysis transforms raw inputs into “data artifacts” — figures, tables, databases, and other concrete outputs. But traditional publication workflows often break this chain of reproducibility. Even when the underlying analysis is reproducible, the manual assembly of publications — selecting figures, crafting captions, formatting tables – introduces human steps that can't be automated or verified. This means that while individual components might be reproducible, the publication as a whole is not. For an analysis to be truly reproducible, anyone should be able to take the same inputs and generate identical artifacts.

CHALLENGE 2: We shouldn't spend too much time polishing pubs when other scientists can benefit from accessing our results now.

Delays mean missed opportunities for early feedback, preventing others from building on our useful intermediate results sooner. Though our scientists know this, they can still feel pressure to polish extensively before sharing.

SOLUTION: Treat the entire publication as a data artifact of the analysis pipeline ([Figure 1](#)).

Rather than manually assembling components, the publication emerges directly from the computational workflow. This approach ensures end-to-end reproducibility, where every element of the final publication — from data processing to narrative text — is generated through documented, reproducible steps. And notebook pubs make it natural to share work at stages we might not traditionally consider “publication-ready,” even though it may be immediately

valuable to the community. The format sets different expectations for a pub — readers understand they're getting direct access to the scientist's working process, complete with its natural progression and iterations. This shift in expectations should make it easier for our scientists to share results that are fresh off the keyboard.

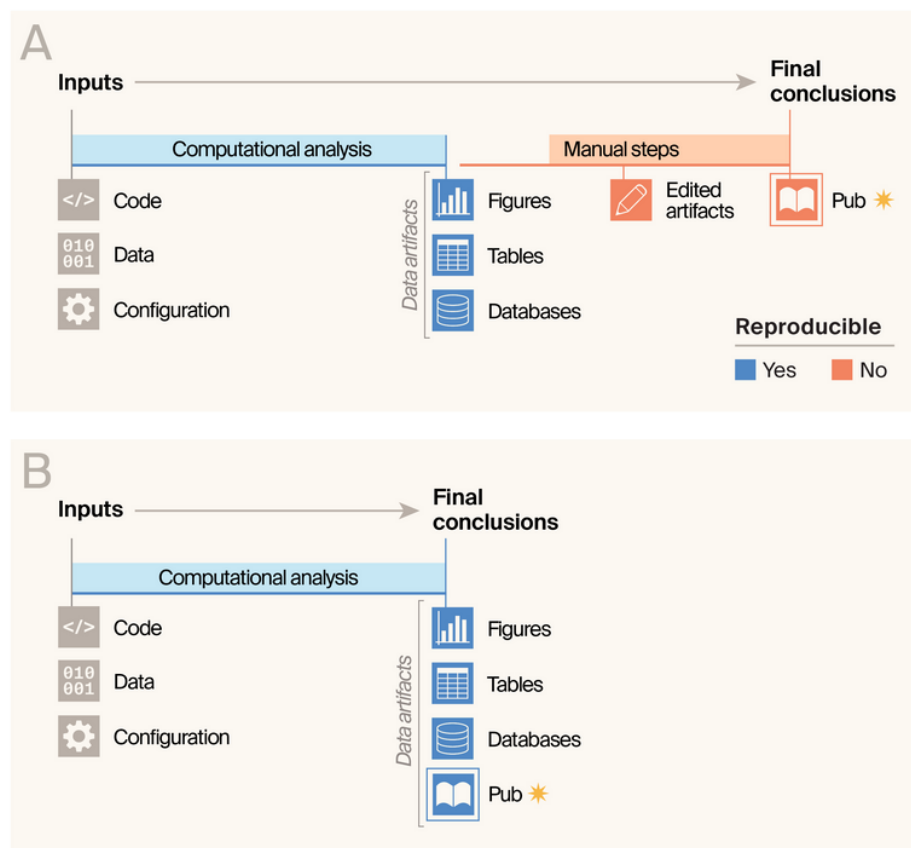


Figure 1. **A visualization comparing traditional versus notebook publication workflows.**

(A) In the traditional workflow, inputs undergo computational analysis to produce data artifacts, including figures, tables, and databases, which are then subjected to manual steps. These manual steps transform the artifacts into edited versions that appear in the final publication.

(B) In the notebook publication workflow, inputs flow directly through computational analysis to create all publication elements as data artifacts. The publication itself becomes another data artifact of the analysis pipeline, eliminating manual editing steps.

Thus, we think notebook pubs should accelerate scientist progress by making knowledge transfer and collaboration more efficient. When methods and analysis are shared in their native, executable format, other researchers can immediately validate results and adapt techniques into their own work. This is also why we chose a GitHub-based approach, as it provides natural pathways for community

engagement — readers can suggest improvements via pull requests, fork to extend analyses, and build upon one another's work at a fast pace. This creates a dynamic and collaborative environment that removes the traditional boundaries between authors and readers.

Publishing workflow

With our scientists' actual workflows in mind, we developed a streamlined publication process that minimizes overhead for researchers while maintaining high standards for scientific communication. The workflow begins when one of our scientists clones our [template GitHub repository](#), which contains a skeleton for their planned analysis, as well as the necessary infrastructure to publish that analysis. By baking our publishing infrastructure into a foundation that underlies our scientists' analyses, each analysis comes equipped with the ability to morph into a publication, allowing the scientist to focus solely on their analysis and narrative.

The scientist can develop their analysis within the notebook template, building upon our pre-configured infrastructure while being able to live-preview how their work will appear as a published document, enabling real-time refinement of both content and presentation. When the analysis is complete, our publishing team reviews the work, does some quality checks, deploys the pub through to our public-facing GitHub Pages site, and links to it from a "stub" pub on our main research site so it can have a DOI, become indexed in Google Scholar, and be searchable alongside other pubs on our site.

By providing standardized infrastructure through a template, we eliminate common technical hurdles while ensuring consistency across publications. The live preview capability allows scientists to iterate quickly, and our publishing team's final review maintains the high standards expected of scientific communications without creating undue burden for our researchers.

TRY IT: Clone our [template](#) and make your own notebook pub.

Under the hood

At the core of our notebook publication system lies Quarto, an open-source scientific and technical publishing system [3]. Quarto serves as the bridge between computational notebooks and polished web publications, handling the complex task of converting notebook content into interactive HTML while preserving code execution, interactive elements, and rich formatting.

When a scientist works within our template, they're actually creating what Quarto calls a "notebook document" — a format that combines executable code, narrative text, and computational outputs. Quarto processes this document through a sophisticated pipeline: it executes all code cells, captures their outputs, and transforms everything into a cohesive HTML publication. This transformation preserves not just the visual elements but also the underlying computational narrative, including code-folding capabilities, interactive visualizations, and detailed execution metadata.

Our template tailors Quarto's functionality with custom styling and navigation elements designed to match Arcadia styling and the way we present our work. We've added supporting pages that provide clear instructions for reproducing the analysis and contributing to the publication. We also include responsive design elements that ensure a seamless reading experience across devices — a crucial feature given that our analytics show more than half of our readers access publications on mobile devices.

The entire publication system operates under what we call a "GitHub umbrella" — each publication exists as a self-contained GitHub repository that handles every aspect of the publication process. Under this model, GitHub serves as a unified platform for managing code, data, and website design. GitHub Actions automates the publication pipeline, GitHub Pages handles hosting, and Giscus provides a commenting system integrated with GitHub Discussions [4]. This approach leverages Git's version control capabilities, allowing us to track changes, manage contributions, and maintain a complete history of the publication's evolution.

The GitHub Actions workflow we've implemented automates the final steps of publication. It runs Quarto's rendering process in a controlled environment, ensures all dependencies are properly managed, and deploys the resulting

website to GitHub Pages. This automation not only guarantees consistency across publications but also maintains the reproducibility chain — from raw data to published results, every step is documented and automated.

Weigh in!

One major goal of this publishing experiment is to engage more deeply with our community. By reducing the lag between discovery and publication, notebook pubs create opportunities for more dynamic scientific discourse. When readers can access our work while it's still actively developing, they become potential contributors rather than just passive consumers. This shift is further enabled by end-to-end reproducibility — readers not only see our results, but can immediately build upon them, with confidence that they can replicate our environment and extend our analyses. The entire publication exists as a living, version-controlled repository where every element — from data to code to narrative — is accessible and modifiable. Whether through comments via Giscus, suggested modifications through pull requests, or full-fledged collaborative extensions, we welcome engagement at any level. Each publication is equipped with instructions for reproducing, and we're hopeful that our standardized infrastructure makes it straightforward to fork and extend our work. We believe this approach not only accelerates individual research efforts but helps build a more collaborative scientific community — one where the traditional boundaries between authors and readers blur, replaced by a network of researchers building on each other's work in near-real time.

The experiment has begun!

Alongside this commentary, we've released **our first notebook pub**, which you can read (and engage with) [here](#).

What's next?

Many of our scientists are hard at work trying out this new format.

Our major next step will be to host notebook pubs directly on our publication platform. We're in the process of upgrading to the newest version of PubPub, which is much more flexible and could accommodate this new format with more development work. We'd especially love to find a way to make code directly executable from within the pub itself, without requiring someone to separately clone or fork the GitHub repo.

And we'd especially like to hear from you — what would make notebook pubs more useful for you, either as someone trying reproduce our work or perhaps as someone interested in sharing their own?

Additional methods

We used ChatGPT to help write code. We used Claude to help write code, suggest wording ideas which we then selectively incorporated, write original text that we edited, rearrange text we provided to fit one of our templates, expand on a summary we provided and then edit the resulting text, and help clarify and streamline text that we wrote.

Contributors (A–Z)

- **Wasim Amiri:** Validation
- **Prachee Avasthi:** Supervision
- **Audrey Bell:** Visualization
- **Brae M. Bigge:** Supervision
- **Keith Cheveralls:** Validation
- **Megan L. Hochstrasser:** Critical Feedback, Editing, Methodology, Supervision
- **Evan Kiefl:** Conceptualization, Methodology, Software, Visualization, Writing
- **Robert Roth:** Critical Feedback, Methodology, Resources

- **Ubadah Sabbagh:** Resources, Writing
- **Ryan York:** Conceptualization, Critical Feedback

References

1. Avasthi P, Hochstrasser ML, Roth R. (2024). The experiment continues: Arcadia publishing 2.0. <https://doi.org/10.57844/arcadia-1c73-8cb5>
2. Avasthi P, Hochstrasser ML, Roth R. (2024). Early update on Arcadia publishing 2.0: Scientists are in charge, speed is an issue. <https://doi.org/10.57844/arcadia-2a89-51c1>
3. Allaire J, Teague C, Xie Y, Dervieux C. (2022). Quarto. <https://doi.org/10.5281/zenodo.5960048>
4. <https://github.com/giscus/giscus>