

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Interactive Version

- [Living Preprint](#)

Code

- [Technical Screening](#)
- [Submitted Repository](#)

Reproducibility Assets

- [Repository](#)
- [Dataset](#)
- [Living Preprint](#)
- [Container](#)

Moderator: Mathieu Boudreau

Screener(s):

- [@mathieuboudreau](#)

Submitted: 17 June 2025

Published: unpublished

License

Authors of papers retain copyright
 and release the work under a
 Creative Commons Attribution 4.0
 International License ([CC BY 4.0](#))

Toward a woven literature: Open-source infrastructure for reproducible publishing

Agah Karakuzu^{1,2}

¹ NeuroPoly Lab, Polytechnique Montreal, Montreal, QC, Canada ² Montreal Heart Institute, Montreal, QC, Canada

Disclaimer



This pdf is intended for content registration purposes only! For full access and interactive reading of this publication, please visit [the live version](#).

Introduction

The trajectory of scholarly communication is marked by transformative technological shifts. It began with the meticulously hand-copied manuscripts of medieval scholars. This was followed by the advent of the first academic journals in the 17th century ([C. Andrade, 1965](#)). In the 19th and early 20th centuries, typewritten manuscripts were circulated via postal services. By the late 20th century, the digital age ushered in online platforms that enabled widespread electronic dissemination and archival of research articles. While these developments dramatically increased accessibility, they did not fundamentally alter the structure or format of scientific publishing. Most articles remained static PDFs, detached from the underlying data, code, and computational environments that produced their findings.

Today, open-source technologies and reproducibility-focused initiatives present an opportunity to reimagine the scientific article as an executable, living document. This review situates that opportunity within a broader technical landscape. It argues that scholarly publishing is on the brink of a transformation, one that could re-align publication with the ethos of open science, computational reproducibility, and collaborative progress.

Admonition

A diagnostic analysis, exposing the technical limitations and economic asymmetries of the legacy publishing complex. We trace how the infrastructure meant to serve science has instead come to extract disproportionate value from it, and how this dynamic has delayed the adoption of more open, interactive modalities. A survey of enabling technologies, charting the open-source tools and cloud platforms that allow authors to create interactive, executable “living publications.” We highlight NeuroLibre as a case study for end-to-end reproducibility at scale—demonstrating how code, data, and computational environments can be woven together into a cohesive and shareable research object. A procedural outlook, exploring how distributed peer review, workflow modularization, and new incentive models can support a more transparent and dynamic publishing ecosystem. We introduce the concept of woven literature, building upon Knuth’s original notion of literate programming, to describe publications where prose, computation, and verification not only coexist but evolve together as integrated components.

Admonition

What are the resource boundaries of browser-based execution environments? How can large datasets be reproducibly integrated without overwhelming readers or computational platforms? When should upstream, resource-intensive workflows transition to lighter, reader-facing phases of analysis? How can emerging tools ensure complete traceability between raw inputs and final results? Which novel reuse patterns become possible through these advanced publishing technologies? By addressing these questions, this review bridges the gap between theoretical potential and practical implementation of next-generation scholarly communication systems.

24

25 *When the elm strangles the vine: A broken equation*

26 The role of academic publishers in advancing scientific progress is undeniable. For centuries,
27 researchers and publishers have shared a mutually beneficial relationship. This partnership is
28 symbolically captured in Elsevier's logo, which features an old man, representing the scientist,
29 harvesting grapes, a metaphor for knowledge, from a vine entwined around an elm tree, which
30 represents the publisher. The accompanying motto, Non Solus ("not alone"), emphasizes the
31 collaborative nature of scientific discovery (Schlüter & Vinken, 1997).

32 The symbiosis between the elm tree and the grapevine is multi-layered and has been cultivated
33 since Ancient Greece (Fuentes-Utrilla et al., 2004). The elm provides structural support for the
34 vine, while its canopy helps regulate the microclimate, improving grape quality. During droughts,
35 the tree may even share its sap with the vine, helping it survive under adverse conditions.
36 Elsevier's logo, though a poetic representation of what science expects from publishers, turns
37 out to be botanically inaccurate: without pruning, the depicted elm would reduce grape yield
38 and undermine its potential to support the vine through sap sharing (Heybroek, 2015).

39 More critically, this symbolism has become practically reversed: what began as collaboration
40 has tilted toward commodification. Instead of publishers sustaining the growth of science, it
41 is now the scientific enterprise at scale that fuels the commercial success of a few dominant
42 publishers. The primary beneficiaries of the advances in research dissemination have been a
43 few dominant publishers, whose profit margins (up to 40-50%) have outpaced even those of
44 the most successful high-tech companies (Larivière et al., 2015), revealing a disproportion
45 between their economic gains and the actual value they add to scholarly communication.

46 *To begin, this discrepancy can be examined through its numerical dimensions:* The real cost of
47 hosting a PDF and registering it as scholarly content is estimated to be around \$2.71 (Katz et
48 al., 2019), assuming an open-access online journal that publishes around 300 articles annually
49 (e.g., the Journal of Open Source Software, JOSS (Smith et al., 2018)). On the other hand,
50 article processing charges (APCs) of a major publisher can reach a staggering \$12,000, with
51 an average of \$3.3k across nearly 14,000 publications per year (Van Noorden, 2013). Notably,
52 one such publisher has claimed that the internal cost of publishing an open-access article
53 ranges between \$30,000 and \$40,000 (Van Noorden, 2013). This nearly 4,500-fold average
54 disparity, which rises to as much as 14,500-fold according to some claims, between the actual
55 cost of \$2.71 and the brand-inflated cost of making an article publicly accessible does not
56 correspond to a proportional increase in scientific impact (Maddi & Sapinho, 2021).

57 *If not delivering impact, is this level of spending at least contributing to the evolution of*
58 *scientific communication?* Unfortunately, an equally pressing concern lies in the underutilization
59 of even basic technological advancements. A telling example is the so-called Continuous Article
60 Publishing model (CAP), introduced in the early 2000s. Here, the term *continuous* refers to
61 the practice of releasing accepted manuscripts online as they are ready, rather than waiting
62 for complete issues to be compiled. Remarkably, some journals are still in the process of
63 transitioning to CAP more than two decades later. This slow adoption underscores the inertia
64 of legacy publishing systems, which struggle to implement even modest procedural changes,

65 let alone the infrastructural transformation needed to support truly modern and integrated
66 forms of scholarly communication (DuPre et al., 2022).

67 As a result of the inertia embedded in legacy publishing systems, digital publication often
68 remains structurally tied to the logic of print. Publications remain static, disconnected from
69 the data, code, and computational environments that would allow them to serve as dynamic
70 components of an evolving research ecosystem. In signal processing terms, *the body of*
71 *knowledge is sampled but never reconstructed*: research exists as discrete outputs without the
72 connective infrastructure for synthesis, independent replication, or reuse.

73 Encased in this digital shell modeled on print-era conventions, academic papers continue to
74 function as isolated artifacts. The literature becomes increasingly bibliographic, where citations
75 serve more as symbolic gestures than as functional links between interoperable knowledge
76 objects. Sticking with the signal processing analogy, *CAP is only about increasing the sampling*
77 *rate* of a bibliographic literature, primarily serving to accelerate publishers' revenue streams,
78 falling far short of the infrastructural leap required for Continuous Science (Cockett, 2024).

79 The concept of Continuous Science represents a fundamental modernization of research dis-
80 semination. It promotes a truly continuous digital medium in which articles are no longer
81 isolated relics but integrated research objects (DuPre et al., 2022). Returning to the signal
82 processing analogy, this is not merely about increasing the sampling rate. It is about *recon-*
83 *structing a coherent and interoperable knowledge system*. Such a system would enhance the
84 signal-to-noise ratio of scientific communication by enabling reproducibility, synthesis, and
85 meaningful connectivity across research outputs.

86 Before turning to the elements of modern solutions that can enable true continuity in open
87 and reproducible scientific knowledge, it is worth considering another perspective on legacy
88 publishing in light of recent advances in generative artificial intelligence (GenAI).

89 ***Legacy publishing can't keep its footing in a GenAI world***

90 Even a decade before large language models (LLMs) made their sweeping entrance into scientific
91 prose (Meyer et al., 2023), the oligopolistic nature of the publishing ecosystem had already
92 been described as the most profitable obsolete technology in history (Schmitt, 2014).

93 Against this backdrop of stagnation, the pace of innovation in AI has been nothing short of
94 astonishing. The second quarter of 2025, just four days before the writing of this manuscript,
95 witnessed the introduction of ARC-AGI-2, a new benchmark in the Abstraction and Reasoning
96 Corpus for Artificial General Intelligence (Chollet et al., 2025): <https://arcprize.org>. This
97 update was prompted by a rapid rise in top performance on the previous benchmark, which
98 increased from 34% in early 2024 to 88% by the end of the year (for the latest leaderboard see
99 (Community, 2025)). For comparison, members of the general public typically score around
100 77%, while STEM graduates score approximately 98%.

101 As LLMs approach expert-level proficiency in generating coherent and convincing prose so
102 rapidly that we need to move the goalpost in favor of humans, it becomes increasingly important
103 to ask: **what remains of a publication when we remove the prose** (Stikov, 2022)?

104 Even twelve years before the initial public release of advanced AI agents, this challenge
105 had already been articulated by (Donoho, 2010): "*An article about computational results is*
106 *advertising, not scholarship. The actual scholarship is the full software environment, code*
107 *and data, that produced the result.*" The legacy publishing system is far too outdated to be
108 retrofitted into an ecosystem where papers serve as more than advertisements.

109 It is only fair to retire the legacy publishing system with a new acronym: **BOOMER** (Barely
110 Operational and Obsolete Manuscript Evaluation Rituals). Perhaps it is time to move toward
111 something more relevant and forward-looking for the century we are in.

112 *From BOOMER to next-gen publishing*

113 Science is what we understand well enough to explain to a computer. Art is everything else we
114 do. [...] People think that science is the art of geniuses, but the actual reality is the opposite,
115 just many people doing things that build on each other, like a wall of mini stones.

116 – Donald Knuth ([Donald E. Knuth & Shustek, 2021](#))

117 Scientific progress is often portrayed as the product of isolated breakthroughs by exceptional
118 individuals. Yet as Knuth reminds us, its true foundation is incremental and collective, where
119 people do small, explainable contributions that build on each other. The BOOMER paradigm
120 stands in the way of this accumulative process, perpetuating a format in which knowledge
121 is trapped in static, monolithic documents, divorced from the computational processes and
122 materials that generated it.

123 Moving beyond this paradigm first requires a return to a more foundational principle that science
124 is what we understand well enough to explain to a computer. This is not merely a technical
125 assertion. It is a claim about findability, accessibility, interoperability, and reproducibility, i.e.,
126 the FAIRness of ideas ([Wilkinson et al., 2016](#)). It is also the central premise of Knuth's concept
127 of literate programming ([D. E. Knuth, 1984](#)), in which the construction of a program is guided
128 not just by what it does, but by how clearly its purpose and logic can be conveyed to others.

129 Literate programming proposes that *narrative and computation should not simply coexist, but*
130 *be intricately woven together* ([D. E. Knuth, 1984](#)). It allows a program to function both
131 as a computational artifact and as a document that communicates scientific intent. Even
132 though it has not been proven effective for professional software engineering ([Lethbridge et al.,](#)
133 [2003](#)), it has become one of the most preferred approaches for end users who code as part of
134 scientific analysis, also known as end-user software engineering ([Ko et al., 2011](#)). For example,
135 a neuroimaging researcher using literate programming tools, such as Jupyter Notebooks ([Perez](#)
136 [& Granger, 2007](#)), to preprocess MRI data is engaging in end-user programming. In the 21st
137 century where computation has become an indispensable part of almost any scientific discipline,
138 literate programming tools boost exploratory analyses by binding prose, code, data, inputs,
139 and outputs into a single space of reasoning ([Kery et al., 2018](#)). Electronic lab notebooks
140 offer a powerful example of such documentation, capturing the narrative of a researcher's data
141 exploration to serve as a valuable asset for reproducibility ([Klokmoose & Zander, 2010](#)).

142 This approach to reproducible documentation naturally informs the design of modern scientific
143 publishing. Inspired by Knuth's concept of literate programming, one can introduce the idea
144 of a **woven literature**. In woven literature, narrative, data, code, and computational runtime
145 are not post-hoc entanglements, but are stitched together at the source of each individual
146 publication. Each thread maintains its integrity, yet contributes to a fabric that is more than
147 the sum of its parts.

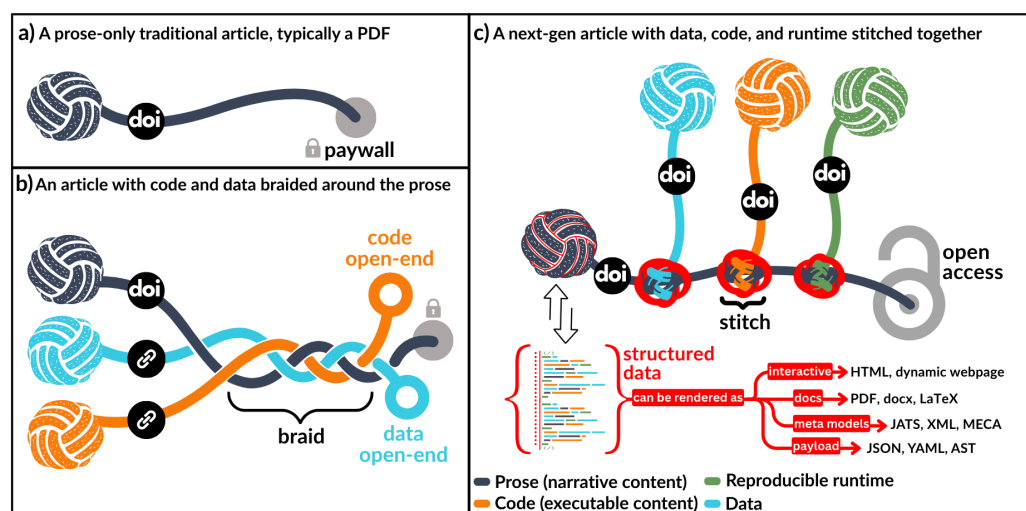


Figure 1: Evolution of article architecture. (a) Traditional prose-only articles are static documents, typically behind paywalls, containing only narrative content. (b) Current articles with code and data availability statements connect to external resources through fragile hyperlinks that are vulnerable to reference rot, shown as braided strands with open ends. (c) Next-generation articles integrate code, data, and runtime environments as core structural components, creating a unified scholarly object. The structured data foundation enables multi-format rendering and serves as the building block for interconnected research networks.

148 This shift is illustrated in **Figure 1**. A traditional static article (Figure 1a) presents a self-
 149 contained and static document, with a narrative frozen at the time of publication. While it
 150 includes references and a DOI that link it to the broader literature, it lacks open and embedded
 151 access to the data, code, or computational context that produced its findings. Even when
 152 such artifacts are available (Figure 1b), they are typically hosted externally and linked as
 153 afterthoughts, which 20% of the time leads to reference rot (Klein et al., 2014). A next-gen
 154 article (Figure 1c), by contrast, embodies the principles of literate programming. It stitches
 155 together code, data, runtime, and narrative into a cohesive structure. Moreover, representing
 156 the article's content as structured data enables seamless transclusion of its components into
 157 other documents across different platforms. This way, the article becomes not just a description
 158 of research but a functional expression of it.

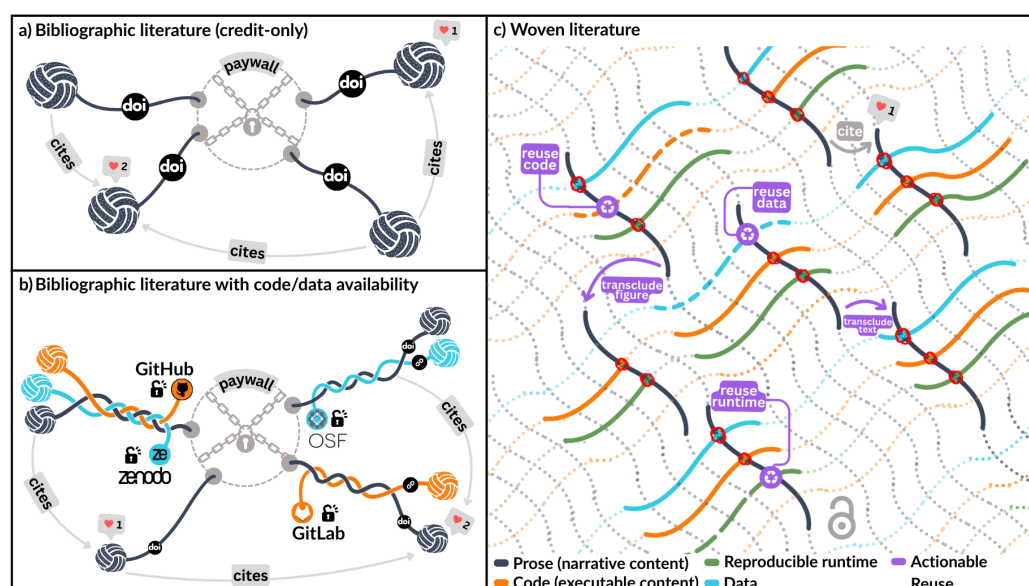


Figure 2: From citation networks to research webs. (a) Traditional bibliographic literature forms simple citation chains between narrative publications, where papers function primarily as units of academic credit. (b) Enhanced bibliographic approaches add peripheral connections to code and data repositories through platforms like GitHub, GitLab, OSF, and Zenodo, but these remain vulnerable to link decay and disconnection. (c) Woven literature creates dense, multi-dimensional networks where research components form actionable connections that enable various reuse patterns such as transclusion (direct insertion of content from one article into another), longitudinal extension as new data becomes available, horizontal adaptation across different datasets, and cross-pollination of research elements throughout the scholarly ecosystem.

Figure 2 shows how this transformation scales beyond individual articles to the scholarly ecosystem. Bibliographic literature (Figure 2a) supports credit and citation but not actionable reuse. Even when reproducibility artifacts are linked Figure 2b, they remain external and structurally isolated. Woven literature Figure 2c introduces formal, machine-readable relationships among the elements of research.

The conceptualized relationships in Figure 2c, such as reuse data or transclude a figure, unlock several reuse patterns:

1. **Transclusion**, when one article directly inserts content from another article such as a paragraph or a figure,
2. **Longitudinal extension**, when a living article re-executes itself with new data points to extend its findings,
3. **Horizontal adaptation**, when an article uses another as a methodological template to apply computational frameworks to independent datasets for validation across different research contexts,
4. **Cross-pollination**, when articles aggregate and synthesize data or methods from multiple sources to generate novel insights that transcend individual publications.

The reuse patterns transform fragmented scientific publishing into what Knuth envisioned as true composability, where small, well-understood computational and narrative components combine to form increasingly sophisticated knowledge structures (Donald E. Knuth, 1993). Such interlacing enables researchers working transparently and methodically to construct robust foundations for discovery, with each publication serving as both standalone contribution and building block for future work.

181 In conclusion, woven literature converts the scientific record from a static collection of isolated
182 PDFs into a dynamic, interconnected infrastructure where cumulative knowledge emerges
183 through systematic reuse and extension of validated components. The following subsection
184 introduces open-source solutions and community efforts making interconnected, executable
185 scholarship a practical reality.

186 ***Open-source pathways to realizing woven literature***

187 NeuroLibre (Karakuzu et al., 2022) represents one of the earliest fully open-source implementa-
188 tions aligned with the principles of woven literature: <https://neurolibre.org>. Developed under
189 the Canadian Open Neuroscience Platform (Harding et al., 2023), it started as a full-fledged
190 reproducible publishing platform for neuroscience by integrating the open-source applications
191 by JOSS (Katz et al., 2018) with the documentation engines of the Jupyter Project (Perez &
192 Granger, 2015). It currently hosts 18 next-gen preprints mostly in the fields of neuroimaging
193 and MRI engineering, with additional submissions actively coming in.

194 Each NeuroLibre publication is deeply interwoven with its reproducibility artifacts, which
195 are automatically archived on Zenodo (<https://zenodo.org>) and served on demand through
196 interactive computational environments. Nearly 60GB of input data is made readily accessible
197 within these containerized reproducible runtimes, enabling real-time exploration and interactivity.

198 Other notable efforts toward modernizing scientific publishing include eLife's *Executable*
199 *Research Articles* (Maciocco et al., 2018) and *Distill* (<https://distill.pub>) (Hohman et al.,
200 2020), both of which are no longer actively maintained. *Wholetale* (<https://wholetale.org>)
201 (Brinckman et al., 2019) offered another model for stitching together reproducibility assets into
202 executable articles, providing a limited set of example publications. *Quarto* (<https://quarto.org>)
203 is an open-source scientific and technical publishing system that supports R, Python, Julia, and
204 JavaScript, allowing users to produce documents, websites, and presentations that combine
205 narrative with executable code. *Observable* (<https://observablehq.com>) provides a reactive
206 notebook environment based in JavaScript, well-suited for building rich, interactive data
207 visualizations and exploratory narratives, though its primary use has been outside formal
208 scientific publishing. In parallel, *Curvenote* (<https://curvenote.com>) (Cockett et al., 2024)
209 has made substantial contributions to next-gen publishing workflows, particularly through its
210 development of Markedly Structured Text (MyST) markdown language (Holdgraf, 2021), a
211 foundational technology for implementing woven literature. Another open-source demonstration
212 of MyST's potential came with you-only-write-thrice, a proof-of-concept workflow that enabled
213 publishing the same content across multiple formats (Sokol & Flach, 2021).

214 While these initiatives differ in openness, scope, and long-term sustainability, NeuroLibre stands
215 out as an actively growing, fully open, and community-driven infrastructure grounded in the
216 principles of open science and reproducibility. Its publication workflow has been made possible
217 by a thriving ecosystem of open-source projects, whose ongoing innovations have not only
218 enabled its core functionality but also steadily expanded its capabilities.

219 The sections below provide a brief review of existing tools that support each thread of a
220 next-gen article, including reproducible runtimes, code and data integration, and editorial
221 management systems that orchestrate the publication process.

222 **Reproducible runtime**

223 One of the main building blocks of NeuroLibre is *BinderHub*, a cloud-native platform that
224 enables users to share reproducible, interactive computing environments directly from code
225 repositories (Freeman & Osherooff, 2016; Ragan-Kelley et al., 2018). Binder streamlines the
226 process of creating containerized environments by reducing complex *Dockerfile* specifications
227 to a small set of configuration files interpreted by language-agnostic buildpacks. This simplicity,
228 combined with its seamless integration with Jupyter Notebooks, has made Binder one of the
229 most widely adopted tools for executable research. As of 2025, the public Binder infrastructure

(<https://mybinder.org>) serves up to 12,000 daily sessions across thousands of repositories (<https://archive.analytics.mybinder.org>), demonstrating its pivotal role in lowering the barrier to reproducible, in-browser scientific computing.

Google Colab is another widely used platform for interactive computing, particularly popular in data science and machine learning education (Bisong, 2019). It allows users to write and execute Python code in a browser-based Jupyter Notebook interface backed by Google's cloud infrastructure. Colab's ease of use, integration with Google Drive, and free access to GPU and TPU resources have made it especially attractive for rapid prototyping, tutorials, and sharing notebooks across diverse user groups.

However, while both BinderHub and Colab offer interactive computational environments, they differ significantly in design philosophy and sustainability model. BinderHub is grounded in open science principles and supports community-led infrastructure. A notable example is 2i2c (<https://2i2c.org>), a nonprofit organization that operates scalable BinderHub services for research and education, exemplifying how interactive computing can be delivered sustainably through a community-driven approach. In contrast, Colab is a freemium product provided by a commercial entity, with priorities shaped by business models rather than academic interests.

BinderHub is fully open source and purpose-built to support reproducible science, emphasizing long-term preservation and transparency. It leverages the Reproducible Execution Environment Specification (REES) (Contributors., 2025), which uses configuration files like *environment.yml*, *requirements.txt*, or *install.R* to explicitly declare and version-control runtime dependencies. These specifications are used to build Docker images that can be hosted in private registries and archived, providing a robust foundation for sustained reproducibility. This model aligns strongly with the goals of scientific publishing, where preserving the integrity and reusability of computational environments is essential.

In contrast, Colab's environments are transient and subject to changes in the underlying base images, which may lead to inconsistencies in results over time. While Colab offers generous access to GPU hardware out of the box (at least as of 2025), BinderHub deployments can be configured to integrate with GPU-enabled *Kubernetes* clusters or JupyterHub spawners, offering comparable computational capabilities within an infrastructure designed for reproducibility and archival.

An emerging alternative leverages *WebAssembly* (Haas et al., 2017), allowing code execution directly in the browser without relying on a remote server. *Pyodide*, a Python distribution compiled to WebAssembly, enables this shift by running Python code client-side in a manner similar to JavaScript (Jefferson et al., 2024). JupyterLite builds on Pyodide to offer a lightweight, serverless Jupyter environment that launches instantly in the browser (Ochkov et al., 2022). Here, the user interacts with notebooks whose code is executed locally, eliminating the need for backend infrastructure.

This approach is particularly exciting for the future of interactive scientific publishing. It offers a scalable solution for delivering interactive papers without being constrained by centralized compute resources, making it ideal for publications with modest computational demands and minimal data dependencies. It opens the door to creating rich, Python-based interactive articles akin to those pioneered by <https://distill.pub>, without requiring expertise in web-native programming. However, ensuring long-term preservation of these browser-executed environments remains an open challenge that warrants further exploration in the context of scholarly publishing.

Code and prose

A wide range of open-source tools now support the creation of executable scientific documents, enabling researchers to combine code and scientific prose in a single environment for reproducible and interactive outputs. *R Markdown*, introduced in 2012, laid the groundwork for integrating analysis and narrative, allowing for dynamic document generation across various

formats (Xie et al., 2018). Two years later, *GitBook* (<https://www.gitbook.com>) emerged, popularizing structured, book-like web documentation, albeit not specifically tailored for scientific communication and currently evolved into a hosted commercial platform. In 2016, *Bookdown* extended R Markdown's capabilities, facilitating the creation of books and long-form reports with features like cross-referencing and multi-format output (Xie, 2016).

The landscape of structured, git-backed web content was bridged with the Jupyter ecosystem by *JupyterBook* (Holdgraf, 2021), introduced in 2019 as part of the *Executable Books* (EB) (<https://executablebooks.org>). Leveraging Sphinx (<https://www.sphinx-doc.org>), a Python-based documentation engine, and MyST markdown language, it renders Jupyter Notebooks and *Markdown* files into interactive, publication-grade documents as static *HTML* files. Features such as collapsible code cells, interactive plotting, and live code execution via *Thebe* (Dehay et al., 2016) enhance its suitability for reproducible and reader-friendly content.

The MyST markdown language has been a major driver of the EB's evolution, enabling its documentation engine to expand beyond a Sphinx-based static site generator toward a framework capable of serving documents as dynamic web pages. This transition (see more [here](#)) was supported through collaboration with Curvenote, which contributed a TypeScript library that leverages the *mdast* package from the *UnifiedJS* project (<https://unifiedjs.com>) to represent the entire content of Jupyter Notebooks and Markdown as abstract syntax trees. Building on this development, MyST-MD (<https://mystmd.org>) enables the full content of an article to be expressed as structured data (Figure 1c), supporting programmatic access and seamless transformation into a wide range of themes (e.g., single-page or book-like web layouts) and formats (e.g., PDF, XML, JATS, MECA, DOCX, or LaTeX) for publishing.

This structured data representation not only eliminates the need for complex format conversions in the publishing pipeline but also unlocks richer online functionality. It enables actionable content reuse, transclusion, and semantic understanding of elements such as figures, narrative text, and code cells. As such, MyST-MD represents a significant milestone in the realization of woven literature, and is currently supported by NeuroLibre for dynamically publishing living preprints.

Data

One of the most commonly missing pieces in building a next-gen article is the input data required for executable content to generate expected outputs. During its early development, the Binder project recognized this challenge and initially recommended storing datasets directly within GitHub repositories (Freeman & Osheroff, 2016). However, this was acknowledged as a stopgap measure rather than a viable long-term strategy, given GitHub's limitations as a platform designed primarily for version-controlling source code. Instead, the Binder team advocated for more robust approaches like Dat (Ogden et al., 2017), a distributed protocol for data synchronization and versioning.

To understand the importance of thoughtful data distribution in scientific publishing, it is useful to reflect on how digital media became widely shareable. Technologies like BitTorrent (Pouwelse et al., 2005) revolutionized peer-to-peer file sharing by allowing users to download large files in small pieces from multiple sources, making the process more efficient and resilient. However, while BitTorrent excels at distributing static content, it offers limited support for collaboration or tracking how files evolve, capabilities that are essential for reproducible science.

Version control systems such as Git introduced structured mechanisms for tracking changes in source code across distributed teams. These same ideas have inspired tools for managing datasets, especially in contexts where traceability and reproducibility are essential. For example, git-annex extends Git to support versioning of large files without storing them directly in the repository. DataLad (<https://www.datalad.org>) (Halchenko et al., 2021), built on top of git and git-annex, is widely used in neuroscience for jointly managing code, data, and the links between them. It enables reproducible research workflows that track both inputs and outputs across time and collaborators.

331 Complementing these solutions, NeuroBagel (<https://neurobagel.org>) tackles a different but
332 equally critical problem: metadata standardization and dataset interoperability. By harmonizing
333 dataset descriptors, NeuroBagel enables federated queries across distributed datasets—making
334 it easier for researchers to discover and access compatible data resources (Urchs et al., 2025).

335 A more general-purpose approach to distributed, versioned data is the InterPlanetary File
336 System (IPFS) (Benet, 2014). Like Dat (Ogden et al., 2017), IPFS identifies files using content
337 hashes, ensuring that a given address always resolves to the same version of a file. Based
338 on peer-to-peer protocols similar to BitTorrent, IPFS can serve as a decentralized backbone
339 for hosting static datasets and artifacts. This content-addressable architecture is particularly
340 valuable in reproducible science, where exact snapshots of datasets must be preserved and
341 referenced across time.

342 Another approach, particularly relevant to scientific computing infrastructures, is the CERN
343 Virtual Machine File System (CVMFS) (Bocchi et al., 2021). Originally developed to distribute
344 software in high-energy physics, CVMFS is well-suited to delivering large-scale datasets efficiently
345 across distributed systems, with support for secure repositories. Its caching mechanisms and
346 content-addressable design complement reproducibility goals by ensuring users access consistent
347 versions of data, even at scale.

348 For platforms like NeuroLibre, which deploys executable scientific articles on JupyterHub
349 instances running atop Kubernetes, mounting datasets to individual user sessions introduces
350 unique challenges. Each reader session requires read-only access to the same inputs to ensure
351 consistent reproducibility, while also being isolated and ephemeral.

352 To address this, NeuroLibre recognizes and supports data hosted on repositories like Zenodo,
353 OSF, and DataLad, using a tool called repo2data to automatically fetch and cache these inputs
354 in a reproducible manner. This makes it possible to reproduce results from linked datasets
355 without requiring manual setup. Looking ahead, the next phase of NeuroLibre's infrastructure
356 will adopt more scalable, efficient solutions such as CVMFS, paving the way for a more robust
357 and larger ecosystem of executable research objects.

358 Editorial management and peer review

359 While advances in tools for code, data, and runtime integration have made it more technically
360 feasible than ever to move beyond the BOOMER paradigm, infrastructure alone does not
361 ensure reproducibility. Without careful curation and modern editorial oversight, executable
362 content written in notebooks may still fall short of yielding truly reproducible outputs (Samuel
363 & Mietchen, 2024).

364 To address this gap, NeuroLibre pioneered the integration of a technical screening process
365 into the living preprint publication workflow. This process supports curation by establishing
366 an iterative interaction between the submitting author, a technical screener, and an editorial
367 bot (RoboNeuro) on a GitHub issue. Within this workflow, the runtime environment is verified,
368 reproducibility checks are logged, and any necessary changes are communicated and resolved
369 until the submission passes all technical criteria (see examples [here](#)).

370 The platform and tooling that support this next-gen technical screening workflow were built
371 from a fork of the Open Journals' applications, *Buffy* and *JOSS* (Smith et al., 2018), and were
372 extended with additional components designed to interface with NeuroLibre APIs. Unlike the
373 JOSS workflow, which involves peer review of both the scientific content and the functionality
374 of the associated research software, NeuroLibre's technical screening focuses solely on ensuring
375 that computational outputs are generated as expected. It does not involve scientific evaluation,
376 similar to commonly used preprint services such as [arXiv.org](#) that has posted more than 1
377 million preprints over the last three decades (Bourne et al., 2017).

378 Interestingly, preprints have not only made publicly funded research more accessible, but have
379 also spearheaded the development of new approaches to peer-review, such as eLife's "publish,
380 then review" model (Eisen et al., 2020). This shift reflects a broader recognition that review

should not be a one-time gatekeeping event as in the BOOMER paradigm, but an ongoing, distributed process that involves diverse perspectives and evolves alongside the work itself. It is on this ground that new models and infrastructures are beginning to take shape, aiming to support more transparent, inclusive, and sustained forms of scholarly evaluation.

A key contributor to advancing this procedural transformation has been the Confederation of Open Access Repositories (COAR), which supports the development of aligned open access infrastructure through a global network of universities, funders, and research institutions (Shearer, 2015, 2024). One of its notable initiatives, COAR Notify, proposes a decentralized mechanism to connect research outputs hosted in distributed repositories with external services (such as overlay journals and open peer review platforms) using linked-data notifications (Walk et al., 2020). This effort stands at the intersection of numerous preprint servers (e.g., arXiv, bioRxiv, medRxiv, Zenodo, Research Square, SciFLO, and Center of Open Science, to name a few), and more importantly, preprint review initiatives such as PreReview (https://prereview.org), Peer Community in (PCI, https://peercommunityin.org), eLife (https://elifesciences.org), Plaudit (https://plaudit.pub), SciPost (https://scipost.org), and Peeref (https://www.peeref.com). Additionally, platforms like ResearchHub (https://researchhub.com) are pioneering novel approaches to post-publication peer review through decentralized evaluation mechanisms and introducing ResearchCoin as an innovative initiative to shift incentive structures in favor of researchers. Collectively, these initiatives represent a growing movement to build a more open, interoperable, and participatory scholarly communication ecosystem where peer review and evaluation are no longer tied to the traditional boundaries of legacy publishers (Henriques et al., 2023).

Results

Figure 3 highlights the structural shift from conventional publishing to a reproducible research article. On the left, the static PDF presents frozen text, figures, and code snippets, which offer limited transparency or reusability. On the right, the same study is rendered as a next-gen article on NeuroLibre, where narrative prose is woven together with executable code, openly accessible data, and an embedded runtime (Salo et al., 2022). This format allows readers to interact with the actual software package and learn about its applications in real-time with zero installation. The NiMARE paper is one of 18 living preprints hosted on NeuroLibre that illustrate this paradigm in practice: https://neurolibre.org.

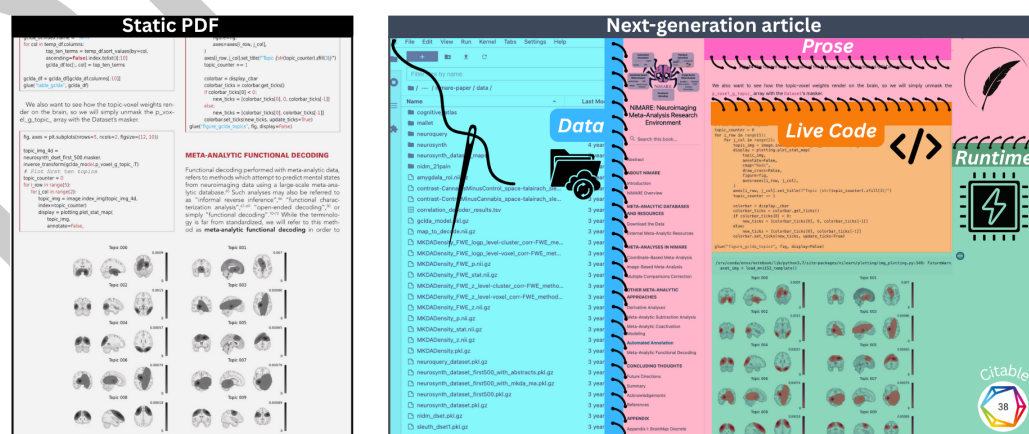


Figure 3: Comparison between a traditional static PDF article (left) and a next-gen reproducible article (right), illustrated using the Neuroimaging Meta-Analysis Research Environment (NiMARE) paper. The next-gen article interlaces prose, live executable code, linked datasets, and computational runtime environments, supporting interactive and transparent scientific communication.

Figure 4 showcases how next-gen publishing on NeuroLibre enables the seamless integration of interactive dashboards as first-class figures. The MRShiny Brain dashboard (<http://shiny-brain.db.neurolibre.org>) offers an interface for exploring structural, perfusion, and metabolic brain imaging data from a normative cohort (Archibald et al., 2024). Built using R Shiny, it empowers readers to navigate multidimensional outcome measures, such as metabolite concentrations and MR thermometry, without being constrained by static layouts or figure limits.

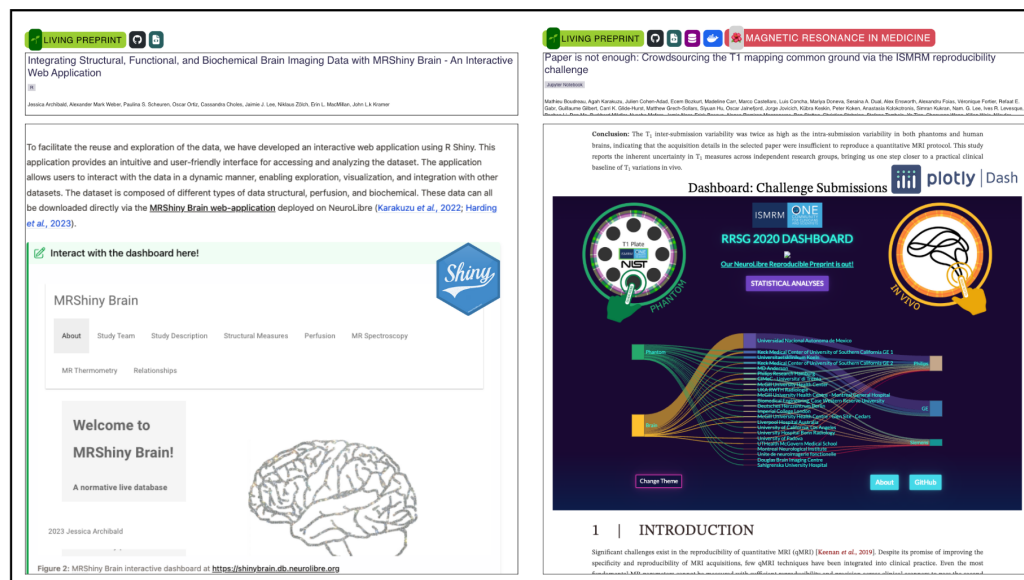


Figure 4: Representative examples of interactive dashboards embedded directly within living preprints hosted by NeuroLibre. The left panel shows MRShiny Brain, an R Shiny dashboard presenting a normative neuroimaging dataset. The right panel features a Plotly Dash-based visualization from the ISMRM T1 Mapping Reproducibility Challenge. Both dashboards are integrated as figures within their respective articles, enabling dynamic exploration of complex results that exceed the limitations of traditional static figures.

On the right, a Plotly Dash dashboard supports the ISMRM T1 Mapping Reproducibility Challenge, visualizing inter- and intra-site variability across phantom and human scans (<https://rrsg2020.db.neurolibre.org>). This web-native figure offers toggles, filters, and drill-down visualizations that would require dozens or even hundreds of static panels in a conventional PDF. These dashboards are deployed on NeuroLibre's dedicated Dokku (<https://dokku.com>) deployment for serving interactive data applications, underscoring the platform's commitment to supporting researchers motivated to communicate complex results through novel, reader-driven interfaces.

Reuse patterns enabled by next-generation publishing infrastructures demonstrate their transformative potential for scientific validation and knowledge synthesis. Longitudinal extension is exemplified by (Boudreau et al., 2024), where the publication evolved continuously through data submissions from multiple sites participating in a reproducibility challenge. The meta-analysis template provided in (Mancini et al., 2020) illustrates the horizontal adaptation pattern, as it was subsequently reused by (Sintu et al., 2025) with an entirely different dataset, enabling systematic validation of methodological robustness across research contexts. Cross-pollination, as defined earlier, can be seen in a proof of concept implementation at <https://agahkarakuzu.github.io/paperception>, yet no real-world examples currently exist.

Discussion

Adoption of next-gen publishing falters from incentives, not interfaces

Despite the maturity and accessibility of technologies enabling next-gen articles, their adoption in mainstream scientific publishing remains limited. The culprit is not an insurmountable learning curve that prevents the broader adoption of such tools. If anything, for the next generation researchers, using tools like Binder and Jupyter Notebooks is as natural as word processors were for those trained in the BOOMER ecosystem. It is worth bearing in mind that this new cohort of scientists has grown up with smartphones and now navigates information fluently using LLMs. For them, the tools of woven literature feel intuitive in a landscape where coding for end-user programming is becoming less a specialized skill and more a facet of AI literacy. In this context, a gen-alpha researcher would aptly describe their fluency as “*vibe coding straight through those interactive plots into something reproducible, no sweat*”.

The real obstacles lie elsewhere. On the publishers’ side, legacy business models tied to static formats continue to dominate, offering little incentive to adopt infrastructures that would undercut their profit margins. On the researchers’ side, entrenched incentive structures and network effects still reward publication in so-called high-impact journals, regardless of the reproducibility or interactivity of the work. Addressing these issues requires not just advocacy, but the development and visibility of alternative venues that demonstrate what modern scholarly communication can look like.

An additional and often overlooked barrier is the difficulty of sustaining platforms like NeuroLibre through traditional scientific funding mechanisms. Despite their potential to transform research communication and reproducibility at scale, infrastructure projects of this nature often fall outside the scope of conventional grant criteria. While programs such as the Chan Zuckerberg Initiative’s open science funding (<https://chanzuckerberg.com>) provide crucial support, there remains a need for broader awareness and recognition within public and national funding systems.

Investments in open, reproducible publishing infrastructure may appear modest in size, but their long-term return in terms of saved time, avoided duplication, and enhanced scientific rigor is substantial. Building sustainable pathways for funding these efforts is essential to realizing the full promise of next-gen publishing.

Defining the edge where a scientific workflow transitions into a next-gen publication

The computational and data demands of next-gen articles vary significantly across research domains. For instance, neuroimaging studies often involve processing large volumes of high-resolution MRI data combined with advanced statistical and machine learning methods. Meanwhile, theoretical papers typically have lower data needs but may require intensive symbolic computation. In contrast, fields such as qualitative social sciences often involve relatively small datasets and minimal computational processing, focusing instead on interpretive analysis.

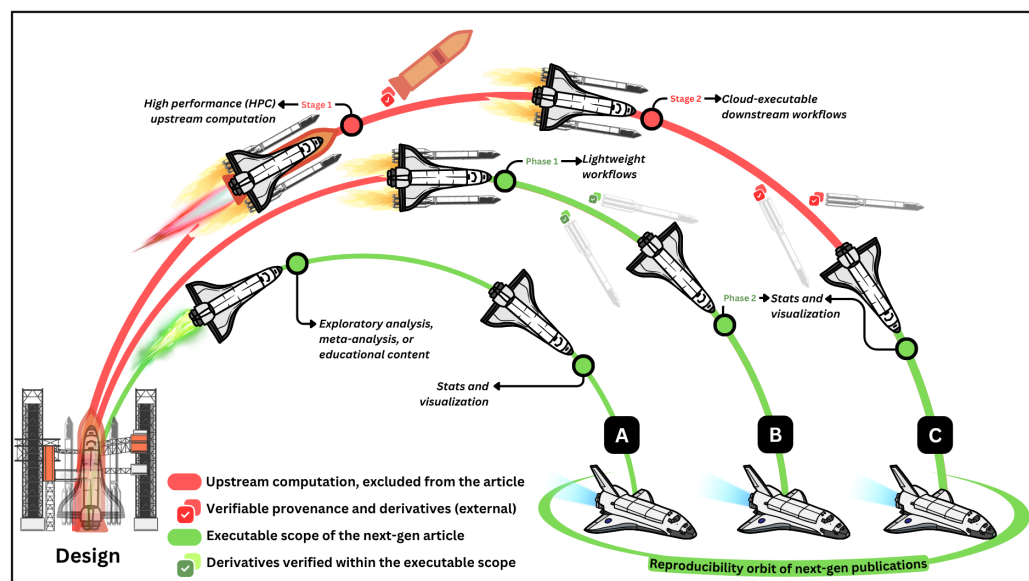


Figure 5: Scientific workflows entering the reproducibility orbit of next-gen publications. Multi-stage tasks are visualized as space missions, where upstream external stages (red arcs) are executed externally and not reproduced in the article, while internal phases (green arcs) fall within the executable scope of the respective next-gen article. Checkmarks indicate verified provenance and derivable outputs from external stages (red) or internal phases (green).

Figure 5 illustrates how scientific workflows enter the reproducibility orbit of next-gen publications by transitioning from external computation stages to internal executable phases. The degree to which a workflow enters this orbit depends on its complexity and structure. Some workflows are fully self-contained, with all operations reproducible in a single session, while others involve modular segments that depend on externally computed stages or pre-generated artifacts that are generated internally.

- **Trajectory A** represents lightweight use cases that do not require batch processing or multi-phase workflow execution logic. All computations, from data ingestion to analysis and visualization, are included within the article and can be executed independently, without strict ordering or dependencies between notebooks. Readers can interact with any part of the content asynchronously, reproducing results in one go without needing to trace interdependent phases.
- **Trajectory B** represents multiple interdependent *phases* that are all technically reproducible within the available computational limits, but differ in practicality for reader interaction. The earlier phases, such as data cleaning, simple model training, or transformation etc., may discourage readers from re-running them interactively. However, these phases are still included in the publication and can be executed if desired. To streamline the reading experience, pre-executed outputs from these initial phases are verified (green checkmarks) and provided as derivatives. Readers are encouraged to engage with the later, more interactive phases, such as analysis, visualization, or interpretation, without being required to reproduce the full pipeline from scratch. This model maintains full transparency and reproducibility, while accommodating asynchronous entry points aligned with reader expectations.
- **Trajectory C** reflects computationally intensive workflows whose upstream *stages* exceed the runtime or resource constraints of next-gen publishing platforms. These workflows often involve multi-step pipelines, such as large-scale HPC preprocessing, simulation, or training procedures, that must be executed externally before publication. Only the final phase, typically involving analysis, visualization, or interpretive synthesis, is included

within the executable scope of the next-gen article.

For example, NeuroLibre currently allocates up to 3GB of RAM and 1 CPU per living print user session, with support for storing up to 4GB of input data. These limits are often sufficient to meet the computational requirements of Trajectory A [Figure 5](#). Representative examples include: i) data-driven exploratory analyses, such as a living preprint by Bellec et al. ([Bellec et al., 2023](#)) showcasing a word feature analysis using scikit-learn ([Kramer, 2016](#)), ii) a science communication analysis for interactive exploration of patent history ([McLean et al., 2024](#)), iii) interactive tutorials covering neuroimaging meta-analyses ([Salo et al., 2022](#)) and quantitative MRI ([Boudreau et al., 2024](#)), and iv) interactive meta-analyses performed focused on myelin imaging ([Mancini et al., 2020](#)) and brain imaging demographics in Quebec ([Sintu et al., 2025](#)).

An example representative of Trajectory B is the living preprint resulting from a multi-center reproducibility challenge organized by the International Society for Magnetic Resonance in Medicine (ISMRM), which evaluated the inter-site reliability of T1 mapping ([Boudreau et al., 2024](#)). In this case, the initial computational phase, such as aligning region-of-interest (ROI) masks and fitting T1 maps, was decoupled from the subsequent statistical analyses, which could be executed within a few minutes.

An example of Trajectory C can be found in a living preprint by Wang et al. ([Wang et al., 2023](#)), which benchmarks the reproducibility of resting-state fMRI denoising strategies across varying preprocessing pipelines. The upstream processing required to generate these results involved running multiple fMRI pipelines under different parameter configurations, an operation that demanded high-performance computing resources beyond the scope of a typical interactive article. As such, only the downstream derivatives, including statistical analysis and visualization of the precomputed outputs, were integrated into the living preprint. This allowed readers to interact with and explore the results reproducibly, even though the computationally intensive preprocessing steps had to be completed externally.

Seamless continuation from upstream workflows to reproducible publishing

Integrating outputs from upstream stages into next-gen articles in a reliable and reproducible way remains an open challenge. Fortunately, platforms like NeuroDesk have begun to address this gap by providing portable, containerized environments capable of running complex neuroimaging workflows ([Renton et al., 2024](#)). NeuroDesk enables reproducible execution and captures the provenance between inputs and outputs, ensuring traceability across computational stages. Similarly, CBRAIN provides a web-based interface to high-performance computing resources, enabling researchers to run large-scale data analyses while preserving metadata and provenance ([Sherif et al., 2014](#)). Another example is BrainLife, focusing on reproducible neuroscience pipelines, offering cloud-based, interactive execution of modular workflows ([Hayashi et al., 2024](#)). More recently, O2S2PARC was introduced as a cloud-based platform with a partial pay-per-use model to support the execution and sharing of multiscale computational models in life sciences ([Guidon et al., 2025](#)).

An integration between NeuroLibre and such platforms could offer a seamless continuation from heavy upstream processing to interactive, downstream communication, bridging the full research workflow that can extend from the study inception to publication ([Karakuzu et al., 2025](#); [Niso et al., 2022](#)). Emerging infrastructures, such as Orvium (<https://orvium.io>), and DeSci Nodes (<https://nodes.desci.com>), further open possibilities for verifiable, tamper-proof records of these computational steps on [blockchain](#), encouraging new models of trust and attribution in scholarly publishing ([Niya et al., 2019](#)).

Conclusion

The ultimate convergence of technical and procedural publishing innovations reviewed in this article will mark a critical turning point in scholarly communication. By integrating

551 reproducibility-aware open infrastructure with distributed models of peer review, scholarly
552 publishing is approaching the replacement of BOOMER's static, one-shot logic with the
553 continuous, collaborative, and verifiable attributes of the woven literature.

554 A Turkish expression, "*ilmek ilmek dokumak*" literally means "*to weave stitch by stitch*" but it
555 is used metaphorically to describe doing something with great care, patience, and attention
556 to detail. This captures the spirit of woven literature: a vision of scholarly publishing built
557 meticulously by interlacing code, data, prose, and peer review into a coherent whole, one
558 next-gen article at a time.

559 As we move forward, this stitch-by-stitch approach offers not just a more reproducible and
560 transparent future, but also one that values the craft of knowledge creation itself—a quality
561 more vital than ever in an era shaped by the rapid rise of generative AI and automated content.
562 The threads are already in motion; what lies ahead is ours to weave. We can choose to remain
563 entangled in the static logic of the BOOMER paradigm, or co-create a knowledge network
564 that evolves through transclusion, modularity, and reproducibility.

565 Acknowledgements

566 I thank Nikola Stikov for insightful discussions, for reading the initial draft of this manuscript,
567 and for providing thoughtful feedback. I am also grateful to the members of the Canadian Open
568 Neuroscience Platform (CONP) Publishing Committee and to the researchers who contributed
569 to the funding and development of NeuroLibre, including, but not limited to, Elizabeth DuPre,
570 Lune Bellec, Jean-Baptiste Poline, Mathieu Boudreau, Patrick Bermudez, Samir Das, and Alan
571 C. Evans. Special thanks to Rowan Cockett and Chris Holdgraf, as well as all members of
572 the Executable Books and Project Jupyter communities for their unwavering commitment to
573 improving the tools that unlocked the next-generation publications.

574 Funding was received from the CONP (<https://conp.ca>), Brain Canada (<https://braincanada.ca>),
575 Courtois Neuromod (<https://www.cneuromod.ca>), Quebec Bio-imaging Network (QBIN),
576 and the Canada First Research Excellence Fund (TransMedTech).

577 Archibald, J., Weber, A. M., Scheuren, P. S., Ortiz, O., Choles, C., Lee, J. J., Zölch, N.,
578 MacMillan, E. L., & Kramer, J. L. k. (2024). Integrating structural, functional, and
579 biochemical brain imaging data with MRShiny brain - an interactive web application.
580 *NeuroLibre Reproducible Preprints*, 29. <https://doi.org/10.55458/neurolibre.00029>

581 Bellec, P., Jbabdi, S., & Craddock, R. C. (2023). Parcellating the parcellation issue - a proof
582 of concept for reproducible analyses using neurolibre. *NeuroLibre Reproducible Preprints*,
583 10. <https://doi.org/10.55458/neurolibre.00010>

584 Benet, J. (2014). IPFS - content addressed, versioned, P2P file system. *CoRR*, *abs/1407.3561*.
585 <http://arxiv.org/abs/1407.3561>

586 Bisong, E. (2019). Google colab. In *Building machine learning and deep learning models on google cloud platform* (pp. 59–64). Apress. https://doi.org/10.1007/978-1-4842-4470-8_7

589 Bocchi, E., Blomer, J., Mosciatti, S., & Valenzuela, A. (2021). CernVM-FS powered container hub. *EPJ Web of Conferences*, 251, 02033. <https://doi.org/10.1051/epjconf/202125102033>

592 Boudreau, M., Karakuzu, A., Cohen-Adad, J., Bozkurt, E., Carr, M., Castellaro, M., Concha, L., Doneva, M., Dual, S. A., Ensworth, A., Foias, A., Fortier, V., Gabr, R. E., Gilbert, G., Glide-Hurst, C. K., Grech-Sollars, M., Hu, S., Jalnefjord, O., Jovicich, J., ... ISMRM Quantitative MR Study Group, the. (2024). Repeat it without me: Crowdsourcing the T1 mapping common ground via the ISMRM reproducibility challenge. *Magnetic Resonance in Medicine*. <https://doi.org/10.1002/mrm.30111>

598 Bourne, P. E., Polka, J. K., Vale, R. D., & Kiley, R. (2017). Ten simple rules to consider regarding preprint submission. *PLoS Computational Biology*, 13(5), e1005473. <https://doi.org/10.1371/journal.pcbi.1005473>

- 601 Brinckman, A., Chard, K., Gaffney, N., Hategan, M., Jones, M. B., Kowalik, K., Kulasekaran, S.,
602 Ludäscher, B., Mecum, B. D., Nabrzyski, J., & Others. (2019). Computing environments
603 for reproducibility: Capturing the “whole tale.” *Future Generation Computer Systems*, 94,
604 854–867. <https://doi.org/10.1016/j.future.2017.12.029>
- 605 C. Andrade, E. N. da. (1965). The birth and early days of the philosophical transactions.
606 *Notes and Records of the Royal Society of London*, 20(1), 9–27. <https://doi.org/10.1098/rsnr.1965.0002>
- 607
- 608 Chollet, F., Knoop, M., Kamradt, G., Landers, B., & Pinkard, H. (2025). ARC-AGI-2: A new
609 challenge for frontier AI reasoning systems. In *arXiv [cs.AI]*. <https://doi.org/10.48550/ARXIV.2505.11831>
- 610
- 611 Cockett, R. (2024). *Continuous science: Accelerating the speed of scientific discoveries through*
612 *changes in collaboration*. <https://doi.org/10.62329/wjck7598>
- 613 Cockett, R., Purves, S., Koch, F., & Morrison, M. (2024). Continuous tools for scientific
614 publishing: Using MyST markdown and curvenote to encourage continuous science practices.
615 *Proceedings of the Python in Science Conference*, 121–136. <https://doi.org/10.25080/nkvc9349>
- 616
- 617 Community, A.-A. (2025). *ARCPRIZE leaderboard*. <https://arcprize.org/leaderboard>
- 618 Contributors., P. J. (2025). *The reproducible execution environment specification*. <https://repo2docker.readthedocs.io/en/latest/specification.html>
- 619
- 620 Dehaye, P.-O., Iancu, M., Kohlhase, M., Konovalov, A., Lelièvre, S., Müller, D., Pfeiffer, M.,
621 Rabe, F., Thiéry, N. M., & Wiesing, T. (2016). Interoperability in the OpenDreamKit
622 project: The math-in-the-middle approach. In *Lecture notes in computer science* (pp.
623 117–131). Springer International Publishing. https://doi.org/10.1007/978-3-319-42547-4_9
- 624
- 625 Donoho, D. L. (2010). An invitation to reproducible computational research. *Biostatistics*
626 *(Oxford, England)*, 11(3), 385–388. <https://doi.org/10.1093/biostatistics/kxq028>
- 627 DuPre, E., Holdgraf, C., Karakuzu, A., Tetrel, L., Bellec, P., Stikov, N., & Poline, J.-B. (2022).
628 Beyond advertising: New infrastructures for publishing integrated research objects. *PLoS*
629 *Computational Biology*, 18(1), e1009651. <https://doi.org/10.1371/journal.pcbi.1009651>
- 630 Eisen, M. B., Akhmanova, A., Behrens, T. E., Harper, D. M., Weigel, D., & Zaidi, M.
631 (2020). Implementing a “publish, then review” model of publishing. *eLife*, 9. <https://doi.org/10.7554/eLife.64910>
- 632
- 633 Freeman, J., & Osheroff, A. (2016). Toward publishing reproducible computation with binder.
634 *eLife [Internet]*, 13.
- 635 Fuentes-Utrilla, P., López-Rodríguez, R. A., & Gil, L. (2004). The historical relationship of
636 elms and vines. *Forest Systems*, 13(1), 7–15. <https://doi.org/10.5424/808>
- 637 Guidon, M., Kaiser, D., Iavarone, E., Anderegg, S., Bisgaard, M., Bujard, C., Cassarà, A.,
638 Crespo-Valero, P., Drobuliak, M., Fasse, A., Hrytsuk, Y., Karimi, F., Maiz, O., Neagu,
639 A., Newton, T., Ordonez, J., Oetiker, T., Pascual, I., Querido, J., ... Neufeld, E. (2025).
640 Shaping a collaborative, sustainable, accessible, and reproducible future for computational
641 modeling. In *bioRxiv*. <https://doi.org/10.1101/2025.06.12.656959>
- 642 Haas, A., Rossberg, A., Schuff, D. L., Titzer, B. L., Holman, M., Gohman, D., Wagner, L.,
643 Zakai, A., & Bastien, J. F. (2017, June). Bringing the web up to speed with WebAssembly.
644 *Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design*
645 *and Implementation*. <https://doi.org/10.1145/3062341.3062363>
- 646 Halchenko, Y. O., Meyer, K., Poldrack, B., Solanky, D. S., Wagner, A. S., Gors, J., MacFarlane,
647 D., Pustina, D., Sochat, V., Ghosh, S. S., Mönch, C., Markiewicz, C. J., Waite, L.,
648 Shlyakhter, I., Vega, A. de la, Hayashi, S., Häusler, C. O., Poline, J.-B., Kadelka, T., ...
649 Hanke, M. (2021). DataLad: Distributed system for joint management of code, data, and
650 their relationship. *Journal of Open Source Software*, 6(63), 3262. <https://doi.org/10.21105/joss.03262>
- 651
- 652 Harding, R. J., Bermudez, P., Bernier, A., Beauvais, M., Bellec, P., Hill, S., Karakuzu, A.,
653 Knoppers, B. M., Pavlidis, P., Poline, J.-B., Roskams, J., Stikov, N., Stone, J., Strother,
654 S., Consortium, C., & Evans, A. C. (2023). The canadian open neuroscience platform-an
655 open science framework for the neuroscience community. *PLoS Computational Biology*,

- 19(7), e1011230. <https://doi.org/10.1371/journal.pcbi.1011230>
- Hayashi, S., Caron, B. A., Heinsfeld, A. S., Vinci-Booher, S., McPherson, B., Bullock, D. N., Bertò, G., Niso, G., Hanekamp, S., Levitas, D., Ray, K., MacKenzie, A., Avesani, P., Kitchell, L., Leong, J. K., Nascimento-Silva, F., Koudoro, S., Willis, H., Jolly, J. K., ... Pestilli, F. (2024). Brainlife.io: A decentralized and open-source cloud platform to support neuroscience research. *Nature Methods*, 21(5), 809–813. <https://doi.org/10.1038/s41592-024-02237-2>
- Henriques, S. O., Rzaeva, N., Pinfield, S., & Waltman, L. (2023). Preprint review services: Disrupting the scholarly communication landscape? In *SocArXiv*. <https://doi.org/10.31235/osf.io/8c6xm>
- Heybroek, H. M. (2015). The elm, tree of milk and wine. *IForest : Biogeosciences and Forestry*, 8(2), 181–186. <https://doi.org/10.3832/for1244-007>
- Hohman, F., Conlen, M., Heer, J., & Chau, D. H. P. (2020). Communicating with interactive articles. *Distill*, 5(9), e28. <https://doi.org/10.23915/distill.00028>
- Holdgraf, C. (2021). Extending open standards for scientific communication with jupyter book and MyST markdown. *AGU Fall Meeting Abstracts*, 2021, IN55D–02.
- Jefferson, T., Gregg, C., & Piech, C. (2024). PyodideU: Unlocking python entirely in a browser for CS1. *Proceedings of the 55th ACM Technical Symposium on Computer Science Education v. 1*, 583–589. <https://doi.org/10.1145/3626252.3630913>
- Karakuzu, A., Biswas, L., Cohen-Adad, J., & Stikov, N. (2022). Vendor-neutral sequences and fully transparent workflows improve inter-vendor reproducibility of quantitative MRI. *Magnetic Resonance in Medicine: Official Journal of the Society of Magnetic Resonance in Medicine / Society of Magnetic Resonance in Medicine*, 88(3), 1212–1228. <https://doi.org/10.1002/mrm.29292>
- Karakuzu, A., Blostein, N., Caron, A. V., Boré, A., Rheault, F., Descoteaux, M., & Stikov, N. (2025). Rethinking MRI as a measurement device through modular and portable pipelines. *Magma (New York, N.Y.)*. <https://doi.org/10.1007/s10334-025-01245-3>
- Katz, D. S., Barba, L. A., Niemeyer, K., & Smith, A. M. (2019). Cost models for running an online open journal. In *Journal of Open Source Software Blog*. <https://doi.org/10.59349/g4fz2-1cr36>
- Katz, D. S., Niemeyer, K. E., & Smith, A. M. (2018). Publish your software: Introducing the journal of open source software (JOSS). *Computing in Science & Engineering*, 20(3), 84–88. <https://doi.org/10.1109/mcse.2018.03221930>
- Kery, M. B., Radensky, M., Arya, M., John, B. E., & Myers, B. A. (2018, April). The story in the notebook: Exploratory data science using a literate programming tool. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. <https://doi.org/10.1145/3173574.3173748>
- Klein, M., Van de Sompel, H., Sanderson, R., Shankar, H., Balakireva, L., Zhou, K., & Tobin, R. (2014). Scholarly context not found: One in five articles suffers from reference rot. *PLoS One*, 9(12), e115253. <https://doi.org/10.1371/journal.pone.0115253>
- Klokmose, C. N., & Zander, P.-O. (2010). Rethinking laboratory notebooks. In *Proceedings of COOP 2010* (pp. 119–139). Springer London. https://doi.org/10.1007/978-1-84996-211-7_8
- Knuth, D. E. (1984). Literate programming. *The Computer Journal*, 27(2), 97–111. <https://doi.org/10.1093/comjnl/27.2.97>
- Knuth, Donald E. (1993). Theory and practice. *arXiv [Cs.GL]*. <https://doi.org/10.48550/ARXIV.CS/9301114>
- Knuth, Donald E., & Shustek, L. (2021). Let's not dumb down the history of computer science. *Communications of the ACM*, 64(2), 33–35. <https://doi.org/10.1145/3442377>
- Ko, A. J., Abraham, R., Beckwith, L., Blackwell, A., Burnett, M., Erwig, M., Scaffidi, C., Lawrance, J., Lieberman, H., Myers, B., Rosson, M. B., Rothermel, G., Shaw, M., & Wiedenbeck, S. (2011). The state of the art in end-user software engineering. *ACM Computing Surveys*, 43(3), 1–44. <https://doi.org/10.1145/1922649.1922658>
- Kramer, O. (2016). Scikit-learn. In *Studies in big data* (pp. 45–53). Springer International Publishing. https://doi.org/10.1007/978-3-319-33383-0_5
- Larivière, V., Haustein, S., & Mongeon, P. (2015). The oligopoly of academic publishers in the

- digital era. *PloS One*, 10(6), e0127502. <https://doi.org/10.1371/journal.pone.0127502>
- Lethbridge, T. C., Singer, J., & Forward, A. (2003). How software engineers use documentation: The state of the practice. *IEEE Software*, 20(6), 35–39. <https://doi.org/10.1109/ms.2003.1241364>
- Maciucci, G., Tsang, E., Bentley, N., & Aufreiter, M. (2018). Reproducible document stack: Towards a scalable solution for reproducible articles. In *eLife*. <https://elifesciences.org/labs/b521cf4d/reproducible-document-stack-towards-a-scalable-solution-for-reproducible-articles>
- Maddi, A., & Sapinho, D. (2021). Article processing charges based publications: To which extent the price explains scientific impact? *Proceedings of the ISSI 2021*.
- Mancini, M., Karakuzu, A., Cohen-Adad, J., Cercignani, M., Nichols, T. E., & Stikov, N. (2020). An interactive meta-analysis of MRI biomarkers of myelin. *eLife*, 9. <https://doi.org/10.7554/eLife.61523>
- McLean, E., Abdo, J., Blostein, N., & Stikov, N. (2024). Little science, big science, and beyond: How amateurs shape the scientific landscape. *NeuroLibre Reproducible Preprints*, 31. <https://doi.org/10.55458/neurolibre.00031>
- Meyer, J. G., Urbanowicz, R. J., Martin, P. C. N., O'Connor, K., Li, R., Peng, P.-C., Bright, T. J., Tatonetti, N., Won, K. J., Gonzalez-Hernandez, G., & Moore, J. H. (2023). ChatGPT and large language models in academia: Opportunities and challenges. *BioData Mining*, 16(1), 20. <https://doi.org/10.1186/s13040-023-00339-9>
- Niso, G., Botvinik-Nezer, R., Appelhoff, S., De La Vega, A., Esteban, O., Etzel, J. A., Finc, K., Ganz, M., Gau, R., Halchenko, Y. O., Herholz, P., Karakuzu, A., Keator, D. B., Markiewicz, C. J., Maumet, C., Pernet, C. R., Pestilli, F., Queder, N., Schmitt, T., ... Rieger, J. W. (2022). Open and reproducible neuroimaging: From study inception to publication. *NeuroImage*, 263, 119623. <https://doi.org/10.1016/j.neuroimage.2022.119623>
- Niya, S. R., Pelloni, L., Wulschleger, S., Schaufelbuhl, A., Bocek, T., Rajendran, L., & Stiller, B. (2019, May). A blockchain-based scientific publishing platform. *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. <https://doi.org/10.1109/bloc.2019.8751379>
- Ochkov, V. F., Stevens, A., & Tikhonov, A. I. (2022, April). Jupyter notebook, JupyterLab – integrated environment for STEM education. *2022 VI International Conference on Information Technologies in Engineering Education (Inforino)*. <https://doi.org/10.1109/inforino53888.2022.9782924>
- Ogden, M., McKelvey, K., Madsen, M. B., & Others. (2017). Dat-distributed dataset synchronization and versioning. *Open Science Framework*, 10(2.2). <https://doi.org/10.31219/osf.io/nsv2c>
- Perez, F., & Granger, B. E. (2007). IPython: A system for interactive scientific computing. *Computing in Science & Engineering*, 9(3), 21–29. <https://doi.org/10.1109/mcse.2007.53>
- Perez, F., & Granger, B. E. (2015). Project jupyter: Computational narratives as the engine of collaborative data science. *Retrieved September, 11(207)*, 108.
- Pouwelse, J., Garbacki, P., Epema, D., & Sips, H. (2005). The bittorrent P2P file-sharing system: Measurements and analysis. In *Peer-to-peer systems IV* (pp. 205–216). Springer Berlin Heidelberg. https://doi.org/10.1007/11558989_19
- Ragan-Kelley, B., Willing, C., Akici, F., Lippa, D., Niederhut, D., & Pacer, M. (2018). Binder 2.0-reproducible, interactive, sharable environments for science at scale. *Proceedings of the 17th Python in Science Conference*, 113–120. <https://doi.org/10.25080/majora-4af1f417-011>
- Renton, A. I., Dao, T. T., Johnstone, T., Civier, O., Sullivan, R. P., White, D. J., Lyons, P., Slade, B. M., Abbott, D. F., Amos, T. J., Bollmann, S., Botting, A., Campbell, M. E. J., Chang, J., Close, T. G., Dörig, M., Eckstein, K., Egan, G. F., Evas, S., ... Bollmann, S. (2024). Neurodesk: An accessible, flexible and portable data analysis environment for reproducible neuroimaging. *Nature Methods*, 21(5), 804–808. <https://doi.org/10.1038/s41592-023-02145-x>
- Salo, T., Yarkoni, T., Nichols, T., Poline, J.-B., Bilgel, M., Bottenhorn, K., Jarecka, D., Kent, J., Kimbler, A., Nielson, D., Oudyk, K., Peraza, J., Pérez, A., Reeders, P., Yanes, J., & Laird, A. (2022). NiMARE: Neuroimaging meta-analysis research environment. *NeuroLibre*

- 766 *Reproducible Preprint Server*, 1(1), 7. <https://doi.org/10.55458/neurolibre.00007>
- 767 Samuel, S., & Mietchen, D. (2024). Computational reproducibility of jupyter notebooks from
768 biomedical publications. *GigaScience*, 13. <https://doi.org/10.1093/gigascience/giad113>
- 769 Schlüter, L., & Vinken, P. J. (1997). *The elsevier non solus imprint*. Elsevier Science.
770 <https://cir.nii.ac.jp/crid/1130000796402639488>
- 771 Schmitt, J. (2014). Academic journals: The most profitable obsolete technology in history. In
772 *Huffpost*. https://www.huffpost.com/entry/academic-journals-the-mos_b_6368204
- 773 Shearer, K. (2015). *Aligning repository networks and the confederation of open access*
774 *repositories (COAR)*.
- 775 Shearer, K. (2024). *Publish, review, curate a better model for scholarly publishing*. <https://doi.org/10.20350/DIGITALCSIC/16767>
- 776 Sherif, T., Rioux, P., Rousseau, M.-E., Kassis, N., Beck, N., Adalat, R., Das, S., Glatard,
777 T., & Evans, A. C. (2014). CBRAIN: A web-based, distributed computing platform
778 for collaborative neuroimaging research. *Frontiers in Neuroinformatics*, 8, 54. <https://doi.org/10.3389/fninf.2014.00054>
- 779 Sintu, T., Oyeniran, O., Anazodo, U., Leener, B. D., & Alonso-Ortiz, E. (2025). Representation
780 in brain imaging research: A quebec demographic overview. *NeuroLibre Reproducible*
781 *Preprints*, 35. <https://doi.org/10.55458/neurolibre.00035>
- 782 Smith, A. M., Niemeyer, K. E., Katz, D. S., Barba, L. A., Githinji, G., Gymrek, M., Huff, K. D.,
783 Madan, C. R., Mayes, A. C., Moerman, K. M., Prins, P., Ram, K., Rokem, A., Teal, T. K.,
784 Guimera, R. V., & Vanderplas, J. T. (2018). Journal of open source software (JOSS): Design
785 and first-year review. *PeerJ Preprints*, 4(e147), e147. <https://doi.org/10.7717/peerj-cs.147>
- 786 Sokol, K., & Flach, P. (2021). You only write thrice: Creating documents, computational
787 notebooks and presentations from a single source. In *arXiv [cs.PL]*. <https://doi.org/10.48550/ARXIV.2107.06639>
- 788 Stikov, N. (2022). ChatGPT and galactica are taking scientific papers to their logical conclusion.
789 In *Kantarot*. <https://qantarot.substack.com/p/chatgpt-and-galactica-are-taking>
- 790 Urchs, S., Dai, A., Armoza, J., Jahanpour, A., Bhagwat, N., & Poline, J.-B. (2025). *Neu-*
791 *robagel_api*. Zenodo. <https://doi.org/10.5281/ZENODO.8088140>
- 792 Van Noorden, R. (2013). Open access: The true cost of science publishing. *Nature*, 495(7442),
793 426–429. <https://doi.org/10.1038/495426a>
- 794 Walk, P., Klein, M., Van de Sompel, H., & Shearer, K. (2020). Modeling overlay peer review
795 processes with linked data notifications. *Confederation of Open Access Repositories*.
- 796 Wang, H.-T., Meisler, S. L., Sharmarke, H., Clarke, N., Paugam, F., Gensollen, N., Markiewicz,
797 C. J., Thirion, B., & Bellec, P. (2023). *A reproducible benchmark of resting-state fMRI*
798 *denoising strategies using fMRIPrep and nilearn*. <https://doi.org/10.55458/neurolibre.00012>
- 799 Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J. J., Appleton, G., Axton, M., Baak, A.,
800 Blomberg, N., Boiten, J.-W., Silva Santos, L. B. da, Bourne, P. E., Bouwman, J., Brookes,
801 A. J., Clark, T., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., Evelo, C. T., Finkers,
802 R., ... Mons, B. (2016). The FAIR guiding principles for scientific data management and
803 stewardship. *Scientific Data*, 3, 160018. <https://doi.org/10.1038/sdata.2016.18>
- 804 Xie, Y. (2016). *Bookdown: Authoring books and technical documents with r markdown*.
805 Chapman; Hall. <https://doi.org/10.1201/9781315204963>
- 806 Xie, Y., Allaire, J. J., & Golemund, G. (2018). *R markdown: The definitive guide*. Chapman;
807 Hall/CRC. <https://doi.org/10.1201/9781138359444>