The Executable Standard: A Strategic Plan for the Journal of Executable Neuroscience (JEN)

The Imperative for Executable Science: From Archival to Operational Reproducibility

The foundation of scientific progress rests upon the principle of verifiability. For decades, the peer-reviewed journal article has served as the primary instrument for disseminating findings, with the implicit promise that the methods described within are sufficient for an independent researcher to reproduce the results and, thereby, validate the conclusions. However, in the realm of computational science, and particularly in computational neuroscience, this promise is increasingly unfulfilled. The growing complexity of models, the proliferation of software dependencies, and the inadequacy of static, narrative descriptions of dynamic computational processes have precipitated a well-documented "reproducibility crisis". This crisis is not merely an academic concern; it represents a fundamental inefficiency in the scientific enterprise, wasting resources, slowing progress, and eroding public trust.

Current publishing standards, which have evolved to encourage or mandate the sharing of code and data, address only a superficial aspect of this problem. They ensure that the constituent parts of a study are *available*, but they do not guarantee that they are *functional* or *interoperable*. The burden of reassembling these components, deciphering undocumented code, and resolving complex software dependencies is placed upon the reader, a task so arduous that it is rarely undertaken. This report outlines a strategic plan for the launch of the *Journal of Executable Neuroscience (JEN)*, a new publishing venue predicated on a transformative principle: shifting the standard from archival availability to operational reproducibility. JEN will not merely ask for the pieces of the puzzle; it will require a fully assembled, verifiable, and executable scientific artifact as a condition of publication. This document details the imperative for this new standard, the structure of the journal designed to uphold it, the technical architecture required to support it, and a strategic roadmap to establish it as the new benchmark for excellence in computational neuroscience.

The Anatomy of Irreproducibility: Lessons from Large-Scale Replication Failures

The challenges of reproducibility are not theoretical. Large-scale, systematic efforts to replicate published findings have provided stark, quantitative evidence of the current system's failings. Among the most comprehensive of these is the Reproducibility Project: Cancer Biology (RPCB), a multi-year initiative that sought to independently replicate key experiments from high-impact papers in preclinical cancer research. The project's final outputs, published in *eLife*, offer a sobering diagnosis of the state of scientific reporting and provide a clear mandate for a new approach.

The RPCB's initial goal was to repeat selected experiments from 53 high-profile papers published between 2010 and 2012. From the outset, the project encountered a fundamental barrier: a systemic lack of detail in the original publications. Of the 193 experiments initially targeted for replication, not a single one was described with sufficient clarity for the project team to design a replication protocol without further information. This finding is critical: it reveals that the standard unit of scientific communication—the peer-reviewed paper—is fundamentally inadequate for conveying the necessary information to reproduce the work it describes. The process of replication was thus transformed into a forensic exercise, requiring extensive communication with the original authors to fill in the gaps. This initial phase of protocol design took far longer and consumed more resources than anticipated, a direct consequence of insufficient documentation in the source material.

These logistical hurdles had a profound impact on the project's scope. The significant, unbudgeted effort required to simply determine *how* an experiment was performed meant that the project could ultimately only attempt to replicate 50 experiments from 23 papers, a fraction of the original target. This demonstrates that irreproducibility imposes a tangible and substantial economic cost on the scientific community. The resources expended on reverse-engineering methods are resources diverted from novel discovery. This economic inefficiency was also observed in the Brazilian Reproducibility Initiative, a similar project aiming to assess the replicability of biomedical science in Brazil. In that effort, protocols were deliberately reconstructed based solely on the information contained in the published articles to simulate a "naturalistic" replication attempt, highlighting the commonality of these documentation shortfalls across the global research landscape.

Beyond the logistical challenges, the scientific outcomes of the RPCB were equally concerning. When the replication experiments were successfully conducted, the results often differed dramatically from the original findings. A key metric, the effect size, was found to be substantially smaller in the replications. The median effect size observed in the replication studies was 85% smaller than that reported in the original experiments. Furthermore, in 92% of the cases where a positive effect was found, the replication effect size was smaller than the original. This phenomenon of "effect size shrinkage" suggests that the current publishing system, which lacks a built-in, immediate mechanism for verification, may systematically favor and amplify initial, dramatic findings that are not robust. A system that requires push-button reproducibility from the outset could serve as a powerful corrective, ensuring that published claims are more closely aligned with the true strength of the evidence. The RPCB's conclusion is unambiguous: there remains "considerable scope for improving the reproducibility of preclinical research". The project serves as a definitive case study, demonstrating that the current model of scientific reporting is failing to meet the basic requirement of enabling verification, leading to wasted resources and potentially inflated scientific claims.

The Limits of Current Policy: The Gap Between Code Availability and Verifiability

In response to the growing awareness of the reproducibility crisis, many leading journals in computational and life sciences have implemented policies requiring authors to make their code and data available. These policies represent a significant and positive cultural shift toward greater transparency. However, a closer examination reveals that they primarily enforce *archival* reproducibility rather than *operational* reproducibility, leaving a critical gap between the availability of research artifacts and their actual usability.

Leading publishers like Springer Nature, whose portfolio includes the *Nature* journals, now have a standard research code policy. This policy requires a "Code Availability Statement" in all original research articles where new code is central to the findings. Authors are strongly encouraged to deposit their code in public repositories and obtain a persistent identifier, with the policy noting that a simple link to a GitHub repository is insufficient for long-term preservation. Similarly, journals such as *PLOS Computational Biology* mandate that all author-generated code essential to the findings be made publicly available without restriction upon publication. In the field of neuroscience, the *Journal of Neuroscience* has a policy on computer code that recommends depositing new code in a suitable repository like ModelDB or GitHub and requires a "Code Accessibility" statement in the methods section detailing how the code can be accessed.

These policies successfully address the first barrier to reproducibility: locating the source code and data. They move the community away from the historically common but unhelpful "code available upon request" standard. However, they stop short of addressing the more formidable technical challenges that prevent the code from being executed. The problem of "dependency hell"—the complex web of specific software libraries, operating systems, and hardware configurations required to run a piece of code—is left entirely for the reader to solve. The process of installing these dependencies, resolving version conflicts, and writing the necessary "glue code" to connect different scripts can be a monumental task, often exceeding the effort required to understand the original scientific concepts.

This gap between availability and executability creates a "reproducibility façade." Journals can signal their commitment to open science by displaying "Data Available" or "Code Available" badges on articles, and authors can fulfill these requirements by uploading a zip file or a link to a repository. Yet, the underlying research may remain practically irreproducible for anyone without the time, resources, and highly specific expertise to reconstruct the original computational environment. The current policies solve the problem of *finding* the artifacts but do not solve the problem of *using* them. They shift the burden of verification from the author and publisher to the broader scientific community, where it remains largely an unfunded and unrewarded activity.

The vision for the *Journal of Executable Neuroscience* is to close this gap. JEN's core premise is that the responsibility for demonstrating executability lies with the author at the time of submission. By mandating a self-contained, containerized "reproducibility bundle" that can be run with a single command, JEN moves beyond the façade of availability to a new standard of guaranteed, operational verifiability. This approach operationalizes the "five pillars of reproducible computational research"—literate programming, version control, compute environment control, persistent data sharing, and documentation—by packaging them into a single, indivisible unit that is the cornerstone of the publication itself.

The Journal of Executable Neuroscience: A New Publishing Model

The *Journal of Executable Neuroscience (JEN)* is founded on a re-conceptualization of the scientific article. It moves beyond the static, narrative-based PDF to a dynamic, interactive, and verifiable format: the Executable Research Article (ERA). The defining feature of an ERA is not merely the presence of supplementary code, but the mandatory submission of a **Reproducibility Bundle**—a self-contained, computationally complete representation of the research. This bundle transforms the article from a description of an experiment into the experiment itself, ready to be re-executed, inspected, and extended by reviewers and readers.

This section details the specification of this bundle, outlines a taxonomy of article types designed to foster a culture of verification, and introduces a novel, dual-track peer-review process that makes computational reproducibility a non-negotiable prerequisite for publication.

Defining the Executable Research Article (ERA) and the Reproducibility Bundle

An Executable Research Article published in JEN is a scholarly manuscript whose primary evidentiary support is its accompanying Reproducibility Bundle. This concept builds upon a rich history of efforts to create more dynamic forms of scientific communication, including "executable papers" and "executable research compendia". While previous initiatives have often treated the executable component as a valuable supplement, JEN elevates it to a co-equal status with the manuscript. The bundle is not an appendix; it is the verifiable core of the published work.

The structure of the Reproducibility Bundle is designed to be a practical implementation of the core principles of reproducible computational science. It is a single, archived package (.tar.gz or similar) that contains all the necessary components to recreate the paper's key computational results. By mandating this structure, JEN forces a separation of concerns and a level of documentation that are considered best practices but are rarely enforced.

The **Reproducibility Bundle Specification** comprises five essential components:

1. **Declarative Experiment Configuration:** A human-readable configuration file, such as params.toml or config.yaml. This file must contain all the key parameters of the model and simulation (e.g., learning rates, neuron counts, connectivity parameters). The experiment must be launchable via a single, simple command that references this file (e.g., snn train --config params.toml). This enforces a clean separation of the scientific logic (the code) from the specific experimental parameters, allowing reviewers and readers to easily explore the parameter space without modifying the source code itself.
2. **Data:** The bundle must contain or provide access to all data necessary to run the experiment. For small datasets, the data files should be included directly within the bundle. For larger datasets that are impractical to package, the bundle must include a script and persistent identifiers (e.g., DOIs from repositories like Zenodo or the Open Science Framework) that allow for the automated download of the required data. This ensures long-term accessibility and avoids the problem of broken links or data being moved.
3. **Environment Container:** A Dockerfile or Containerfile that declaratively specifies the complete computational environment. This includes the operating system, system-level libraries, programming language versions (e.g., Python 3.9), and all required software packages with their exact versions (e.g., torch==1.10.1). This component is the technical solution to "dependency hell," as it captures the full software stack in a way that can be reliably rebuilt and executed on any machine with a container runtime, effectively insulating the experiment from the user's local environment.
4. **Raw Run Artifacts:** The direct, unprocessed output files generated by the execution of the primary command. In the context of spiking neural network simulations, this would include files containing raw spike times, membrane potentials, synaptic weights, and simulation logs. These artifacts serve as the ground-truth data from which all figures and tables in the manuscript are generated, creating a transparent and auditable trail from raw simulation output to final published result.

5. **Visualization State:** A configuration file and associated scripts that allow a visualization tool (e.g., snn viz serve) to load the raw run artifacts and programmatically regenerate the exact figures presented in the manuscript. This final component closes the reproducibility loop, linking the visual evidence presented in the paper directly and automatically to the raw output of the verified computational experiment.

A Taxonomy of Executable Contributions

To build a thriving ecosystem around executable science, JEN will publish more than just traditional research articles. The journal will feature a carefully curated set of article types, each designed to reinforce the value of reproducibility and create a self-sustaining cycle of verification and extension.

- **Executable Research Articles (ERAs):** This is the flagship article type for presenting original, novel research in computational neuroscience. Submission of a complete and functional Reproducibility Bundle is a mandatory, non-negotiable condition for acceptance. These articles will form the core content of the journal and serve as the substrate for other article types.
- **Methods & Frameworks:** This category is for papers that introduce new computational methods, software frameworks, models, or analysis tools. The primary deliverable is not just a description of the tool, but a Reproducibility Bundle that serves as a working example. For instance, a paper introducing a new SNN simulator would be accompanied by a bundle that allows the reader to immediately run a benchmark simulation. This dramatically lowers the barrier to adoption and allows the community to quickly verify the utility and correctness of new tools.
- **Replication Reports:** This article type is directly inspired by the successful model of journals like *ReScience C*, which are dedicated to publishing computational replications. A Replication Report in JEN takes the Reproducibility Bundle from a previously published article (either in JEN or another venue) and subjects it to further scrutiny. This could involve a straight re-execution to confirm the original results, an analysis of the model's robustness by systematically varying parameters, or an attempt to apply the model to a new dataset. These reports provide a formal, citable, and peer-reviewed venue for the essential scientific work of verification, which is often disincentivized in traditional journals that prioritize novelty.
- **Perspective Papers:** To foster community dialogue and guide the evolution of the field, JEN will publish essays and opinion pieces on topics central to its mission. These include discussions on best practices for reproducibility, the development of new standards and infrastructure, the ethics of open science, and critiques of existing scientific evaluation systems. These papers provide the intellectual and community-building context for the journal's technical contributions.

This taxonomy creates a powerful feedback loop. Executable Research Articles provide the material for Replication Reports. The findings of Replication Reports, in turn, validate and build confidence in the original work, while also providing feedback that can improve future models. Methods & Frameworks articles provide the tools needed to build the next generation of ERAs. This integrated structure ensures that the journal is not just a collection of individual papers, but a dynamic ecosystem dedicated to advancing robust and verifiable computational neuroscience.

The Dual Peer-Review Process: Scientific and Executable Verification

The peer-review process at JEN is designed to be the guardian of its core principles. It introduces a novel, two-stage system that rigorously assesses both the scientific merit and the computational validity of every submission. This process fundamentally rebalances the role of the reviewer, freeing them from the tedious and often impossible task of debugging an author's code and empowering them to engage with the work as an interactive scientific artifact.

Stage 1: Automated Triage via the CI/CD Pipeline

Upon submission through the journal's portal, the Reproducibility Bundle is not immediately sent to a human editor. Instead, it is routed to an automated Continuous Integration/Continuous Deployment (CI/CD) pipeline. This pipeline acts as an impartial, computational gatekeeper.

1. **Build:** The pipeline first attempts to build the container image from the provided Dockerfile. If the build fails—due to missing dependencies, syntax errors, or other issues—the process halts.
2. **Execute:** If the container builds successfully, the pipeline then executes the single command specified by the author (e.g., snn train). It monitors the execution for a predetermined time limit and resource ceiling.
3. **Report:** If the execution completes successfully (with an exit code of 0), the submission is automatically adorned with a "Computationally Verified" badge and is passed into the human review queue. If the build or execution fails, the submission is automatically returned to the author along with the complete build and execution logs.

This automated first pass is a critical innovation. It ensures that no editor or reviewer time is wasted on submissions that are not computationally sound. It provides immediate, objective, and actionable feedback to authors, shifting the responsibility of ensuring basic executability entirely to them. This automates the laborious initial check that reviewers for journals like *ReScience C* must perform manually, streamlining the entire process.

Stage 2: Human Review

Once a submission has passed the automated triage, it proceeds to a dual-track human review process.

- **Scientific Review:** This track follows the traditional model of peer review. Experts in the relevant subfield of neuroscience are tasked with evaluating the manuscript's scientific contributions. They assess the novelty of the research question, the soundness of the theoretical framework, the appropriateness of the model, the significance of the conclusions, and the clarity of the writing.
- **Executable Review:** This track is unique to JEN. Reviewers assigned to this track are also domain experts, but their primary task is to interact with the now-verified Reproducibility Bundle. Their role is not to *debug* the code—the CI pipeline has already guaranteed it runs—but to *interrogate* the science through the executable artifact. Using the provided visualization tools (e.g., snn viz serve), they will:
    - Verify that the figures generated from the bundle's artifacts perfectly match those in the manuscript.
    - Inspect the raw outputs and intermediate results to gain a deeper understanding of the model's behavior.
    - Critically assess the relationship between the code, the results, and the claims made in the paper.
    - They are encouraged, though not required, to perform simple "sensitivity analyses" by modifying parameters in the configuration file and re-running the experiment to probe the robustness of the findings.

This process transforms the nature of peer review. It moves from a passive act of reading and critique to an active process of scientific exploration and verification. By providing reviewers with a functional, interactive "digital lab," JEN empowers them to engage with the research at a depth that is impossible with a static PDF, ultimately leading to more rigorous and reliable published science.

The Technical Architecture for Verifiable Publishing

To realize the vision of the *Journal of Executable Neuroscience*, a robust and scalable technical infrastructure is paramount. This infrastructure must seamlessly manage the submission of complex computational artifacts, automate their verification, provide an interactive execution environment for readers and reviewers, and ensure their long-term archival and future runnability. This section details the technical blueprint for JEN, evaluating existing technologies to propose a stack that is powerful, sustainable, and aligned with the principles of open science. The architecture is composed of three core components: the submission and CI/CD pipeline, the interactive execution sandbox, and the archival and preservation system.

The Submission and CI/CD Pipeline: The Gateway to Review

The entry point for all content into JEN is a specialized submission portal integrated with a powerful automated verification pipeline. This system is designed to be the first line of quality control, ensuring that only computationally sound submissions reach human editors and reviewers.

- **Submission Portal:** The foundation of the submission system will be a mature, open-source journal management platform. Leading candidates include **Open Journal Systems (OJS)** and **Janeway** , both of which are widely used in academic publishing and offer the flexibility needed for customization. The chosen platform will be extended to handle a two-part submission: the traditional manuscript (in PDF or LaTeX format) and the Reproducibility Bundle (as a .tar.gz archive or a link to a version-controlled repository, such as a specific Git commit hash). The submission form will require authors to specify the single command needed to execute their experiment, which will be passed as metadata to the CI pipeline.
- **Continuous Integration/Continuous Deployment (CI/CD) Pipeline:** The portal will be tightly integrated with a CI/CD service using webhooks. When an author submits a new manuscript, the portal will automatically trigger a workflow on a platform like **GitHub Actions**, **GitLab CI**, or a self-hosted **Jenkins** instance. These tools are the industry standard for automating software building and testing and are perfectly suited for the task of verifying the Reproducibility Bundle.

The CI/CD pipeline workflow will proceed as follows:

1. **Environment Provisioning:** A clean, virtualized runner is spun up.
2. **Artifact Retrieval:** The pipeline checks out the specified Git commit or downloads and unpacks the submitted .tar.gz archive.
3. **Container Build:** The pipeline executes a docker build or podman build command, using the Dockerfile provided within the bundle. This step tests whether the specified environment can be constructed successfully.
4. **Experiment Execution:** If the container builds, the pipeline runs the container, executing the author-specified command (e.g., snn train --config params.toml). The execution is monitored against predefined time and resource limits to prevent runaway processes.
5. **Status Reporting:** The pipeline captures the exit code and all standard output and error

streams from the execution.

  - **On Success (Exit Code 0):** The pipeline reports a "success" status back to the journal portal. The portal then attaches a "Computationally Verified" badge to the submission and moves it into the queue for assignment to a scientific editor.
  - **On Failure (Non-Zero Exit Code or Timeout):** The pipeline reports a "failure" status. The complete build and execution logs are captured and attached to the submission, which is then automatically returned to the author with a notification explaining that the bundle failed verification.

This automated triage system is a cornerstone of JEN's operational model. It enforces a high standard of computational quality before any human effort is expended, creating a highly efficient and scalable review process.

The Execution Sandbox: From Static Paper to Interactive Lab

A central feature of JEN will be a "Run Paper" button on every article page, allowing readers to instantly launch and interact with the published Reproducibility Bundle in a cloud-based environment. This transforms the passive reading experience into an active, hands-on scientific exploration. Several powerful platforms exist to provide this functionality. A comparative analysis is essential to select the optimal solution for JEN.

- **Technology Evaluation:** The primary candidates for powering the execution sandbox are Code Ocean, BinderHub, Whole Tale, and Stencila. Each offers a different balance of features, cost, and alignment with open-source principles.
  - **Code Ocean:** This is a mature, commercial platform that provides "Compute Capsules"—self-contained, executable research objects. Its major strengths are a highly polished user interface, robust enterprise-grade infrastructure, and existing partnerships with major publishers like Springer Nature, which have demonstrated the viability of integrating such a platform into the journal submission workflow. However, adopting Code Ocean would introduce a dependency on a proprietary service and entail licensing costs, which may be substantial. While they offer free tiers for academic labs and open science, a journal-level integration would likely require a commercial agreement.
  - **BinderHub:** BinderHub is the open-source technology that powers the popular mybinder.org service. It is designed specifically to build Docker images from code repositories and launch them as interactive sessions (typically Jupyter notebooks) in the cloud. Its primary advantages are its complete alignment with open-source principles and its strong, active community, including support from organizations like The Turing Way. The main challenge is operational: while mybinder.org is a free public service, its resources are limited and not suitable for the guaranteed performance a journal would require. Therefore, JEN would need to deploy and maintain its own dedicated BinderHub instance, incurring cloud computing and infrastructure management costs.
  - **Whole Tale:** This is an NSF-funded, open-source platform specifically developed for the creation and publication of "Tales"—executable research objects that package code, data, and the computational environment. Architecturally, it shares similarities with BinderHub but places a stronger emphasis on integrating with data repositories and capturing detailed scientific provenance information. Its academic origins and focus on the entire research lifecycle make it an excellent philosophical fit for JEN. As the platform continues to mature, it represents a very strong

candidate.
    - **Stencila:** This is the open-source toolkit that powered *eLife's* Executable Research Articles (ERAs). Stencila's approach focuses on embedding executable code cells directly within the narrative text of an article, rather than packaging a separate, standalone bundle. While this "literate programming" model is different from JEN's "reproducibility bundle" concept, Stencila's experience in deploying containerized execution environments for a publisher provides valuable precedent.
- **Recommendation and Implementation:** The recommended strategy for JEN is to launch with a dedicated, self-hosted **BinderHub** deployment. This approach offers the greatest control, avoids vendor lock-in, and aligns perfectly with the journal's open-source ethos. The operational costs of cloud infrastructure and management are a necessary investment in the journal's core value proposition. In parallel, JEN should pursue a partnership with **Code Ocean**. This could provide a secondary, premium execution option for readers and leverage their expertise in publisher integration. The existence of two independent execution pathways would also enhance the robustness and long-term viability of the journal's interactive features.

Feature	Binder/BinderHub	Whole Tale	Code Ocean	Stencila
Core Concept	On-demand interactive environments from Git repositories	Executable research objects ("Tales") combining data, code, and environment	"Compute Capsules" as self-contained, versioned research packages	Literate programming with executable code cells embedded in the article text
Architecture	Kubernetes, repo2docker, JupyterHub	Extends repo2docker; focus on data integration and provenance capture	Cloud-based platform with proprietary containerization and workflow management	Open-source stack for converting documents (e.g., R Markdown, Jupyter) into web-native ERAs
Primary Use Case	Interactive exploration, teaching, sharing computational narratives	Formal publication of reproducible research artifacts, data analysis	Enterprise and academic research workflows, publisher-integrated code review and execution	Creating interactive figures and in-line results within a traditional article format
Data Handling	Primarily data bundled within the Git repo; limited support for external data	First-class support for registering and integrating data from external repositories	Manages data as separate assets that can be attached to multiple capsules	Data files are packaged with the ERA for use by the embedded code cells
Provenance Capture	Implicit via Git commit hash; no formal scientific provenance model	Explicit capture of computational provenance as part of the "Tale"	Automated result provenance and lineage graph generation	Implicitly captured in the code that generates a specific output
Cost Model (Academic)	Free public service (mybinder.org); self-hosting costs	Free public service; self-hosting option	Free for small academic labs and open science, but	Open-source stack; execution hosting costs

Feature	Binder/BinderHub	Whole Tale	Code Ocean	Stencila
	for dedicated hub	available	enterprise/publisher integration is a paid service	covered by publisher (*eLife*) or self-hosted
Open Source	Yes, BSD-3-Clause License	Yes, Apache 2.0 License	No, proprietary platform built on open-source components	Yes, Apache 2.0 License
Journal Integration	Biorxiv pilot via 2i2c	Designed for integration with data repositories and publishing workflows	Integrated with Springer Nature (*Nature*, *BMC*), *Genome Biology*	Integrated with *eLife*
Suitability for JEN	**High.** Aligns with open-source values but requires dedicated hosting.	**High.** Strong academic alignment and focus on provenance. A mature future option.	**Medium-High.** Polished UX and proven publisher model, but proprietary and costly.	**Low.** Different paradigm (embedded code vs. whole bundle), but provides useful precedent.

Archival and Long-Term Executability: The Digital Rosetta Stone

A core commitment of JEN is to ensure that published research remains not just available, but executable, for a decade or more. This presents a significant challenge, as "software rot"—the process by which software becomes unusable due to changes in its underlying hardware and software dependencies—is a pervasive problem. Containerization provides a powerful solution for short-to-medium-term reproducibility, but a more robust strategy is needed for true long-term preservation.

- **Persistent Archival with DOIs:** The first step is to ensure the bundle itself is preserved and citable. Upon acceptance of a manuscript, its corresponding Reproducibility Bundle will be deposited in a long-term public archive such as **Zenodo** or the **Open Science Framework (OSF)**. This action will automatically mint a persistent Digital Object Identifier (DOI) for the bundle, making it a formally citable, first-class research output, independent of the paper's DOI.
- **Standardizing the Bundle with RO-Crate:** To combat software rot and ensure the bundle remains comprehensible to future technologies, its structure and contents must be described in a machine-readable format. JEN will therefore mandate that every Reproducibility Bundle must be a valid **Research Object Crate (RO-Crate)**. RO-Crate is a lightweight, community-developed standard for packaging research artifacts with rich, structured metadata using JSON-LD.JEN will define and publish a formal **"JEN Reproducibility Bundle" RO-Crate profile**. A profile is a set of rules and conventions that tailor the general RO-Crate standard for a specific use case. The JEN profile will require the ro-crate-metadata.json file at the root of the bundle to explicitly declare and describe the five required components:
    1. The path to the declarative configuration file (params.toml).
    2. A list of data sources, including DOIs for external datasets.
    3. The path to the Dockerfile.

4. A description of the raw run artifacts.
5. The command and configuration for the visualization tool.

This RO-Crate manifest acts as a "Digital Rosetta Stone" for the bundle. Even if Docker becomes obsolete in the future, a digital archivist can read the manifest and understand the precise components of the experiment and their relationships. This structured metadata provides a clear roadmap for migrating the experiment to future emulation or virtualization technologies, making 10+ year executability a plausible goal.

- **Source Code Preservation:** In addition to archiving the bundle, JEN will partner with initiatives like **Software Heritage** to archive the source code of the bundle's contents and, where possible, its open-source dependencies. This provides an additional layer of preservation, safeguarding the fundamental building blocks of the research against repository disappearance or corruption.

Governance, Community, and Sustainable Impact

The success of the *Journal of Executable Neuroscience* will depend on more than its technical infrastructure; it requires a robust organizational structure, deep engagement with its scientific community, and a sustainable model that can drive genuine cultural change. A journal that aims to set a new standard cannot operate in isolation. It must be governed with transparency, build strategic alliances with key institutions, and, most importantly, create a new system of incentives that rewards researchers for producing verifiable and reusable work. This section outlines the governance model, partnership strategy, and the innovative "Executable Citation Index" designed to ensure JEN's long-term viability and impact.

A Community-Centric Governance Model: Learning from JOSS and eLife

The governance of JEN will be designed from the ground up to be transparent, mission-driven, and community-led, drawing inspiration from successful models in open-access and software publishing. The structure will be multi-layered to reflect the journal's dual identity as both a scientific venue and a complex sociotechnical platform.

- **Board of Directors:** The highest level of oversight will be provided by a Board of Directors. This board will be responsible for the strategic direction, financial sustainability, and ethical integrity of the journal. Following the model of *eLife*, which is governed by representatives from its founding funders (Howard Hughes Medical Institute, Max Planck Society, Wellcome) and other external experts , the JEN board will include representatives from key funding partners, leadership from allied research institutions (such as the Allen Institute and EBRAINS), and prominent advocates for open science and reproducibility.
- **Executive Editorial Board:** The scientific leadership of the journal will be managed by a tiered editorial board, a structure proven effective by the *Journal of Open Source Software (JOSS)*. This structure allows for both broad oversight and specialized expertise:
    - An **Editor-in-Chief** will provide overall scientific vision.
    - A small team of **Associate Editors-in-Chief** will manage broad sub-domains of computational neuroscience (e.g., Theoretical & Computational Models, Systems & Circuits, Neuro-inspired AI).
    - A large and diverse board of **Topic Editors** will be responsible for the day-to-day handling of manuscripts, including selecting reviewers and making final editorial

decisions. This distributed model empowers a wide range of community experts and ensures that submissions are handled by editors with deep domain knowledge.

- **Technical Advisory Board:** Recognizing that JEN's infrastructure is a core part of its product, a dedicated Technical Advisory Board will be established. This board, composed of experienced research software engineers (RSEs), infrastructure specialists, and experts in data standards, will provide guidance on the evolution of the journal's technical stack. They will oversee the development of the Reproducibility Bundle specification, advise on the selection of technologies for the execution sandbox, and ensure the long-term viability of the archival strategy. This separation of scientific and technical governance acknowledges that maintaining a cutting-edge publishing platform requires a distinct set of expertise from traditional scientific editing.

The journal will operate under publicly stated principles of transparency and best practice in scholarly publishing, adhering to the guidelines set forth by organizations like the Committee on Publication Ethics (COPE) and the Directory of Open Access Journals (DOAJ).

Forging Strategic Alliances: Building an Ecosystem

To achieve its mission, JEN must be deeply embedded within the computational neuroscience ecosystem. It will actively pursue strategic partnerships with institutions and initiatives that share its commitment to open and reproducible science.

- **Institutional Partners:** Formal partnerships will be sought with world-leading research centers in neuroscience. The **Allen Institute for Brain Science**, with its large-scale data generation and open science mission , **EBRAINS**, with its focus on building a European research infrastructure for brain science, and the **Open Neuromorphic** community are natural allies. These partnerships could involve co-sponsoring workshops on creating reproducible workflows, collaborating on the development of data and model standards that align with the JEN bundle format, and encouraging researchers at these institutions to submit their high-impact work to JEN, thereby lending the journal immediate credibility.
- **Reproducibility Initiatives:** JEN will collaborate closely with existing communities dedicated to improving research practices. This includes aligning the JEN bundle specification with the principles and guides developed by **The Turing Way** , ensuring interoperability with standards like **NeuroML**, and engaging with the community around tools like **Sumatra** that focus on tracking computational provenance. By building on and contributing to these existing efforts, JEN can avoid reinventing the wheel and help solidify a unified set of best practices for the entire field.
- **Funder Partnerships:** A key long-term strategy will be to engage directly with major science funding agencies, including the National Science Foundation (NSF), the European Research Council (ERC), the Wellcome Trust, and the Simons Foundation. The goal is to position JEN as a premier publishing venue for their grantees and to advocate for the JEN Reproducibility Bundle standard to be recognized or required as a key deliverable for funded computational research projects. This alignment would create a powerful incentive for researchers to adopt JEN's standards from the very beginning of their research lifecycle.

The Executable Citation Index: Redefining Research Impact

The single greatest barrier to the widespread adoption of reproducible practices is the academic incentive structure, which overwhelmingly rewards novel publications in high-impact-factor journals while providing little to no formal credit for the time-consuming work of creating robust, reusable, and well-documented software and data artifacts. To overcome this barrier, JEN will introduce its most significant innovation: the **Executable Citation Index**, a new system for measuring and rewarding the impact of a research artifact itself, not just the paper that describes it.

- **Technical Implementation:** The foundation of this index is the ability to uniquely identify and cite the Reproducibility Bundle.
    1. **Assigning DOIs to Bundles:** As described in the archival plan, every published bundle will be deposited in Zenodo, which automatically assigns it a persistent DataCite DOI. This elevates the bundle from "supplementary material" to a first-class, citable research object.
    2. **Mandatory and Granular Citation:** JEN's author guidelines will mandate a new citation practice. Any future work that builds upon, reuses, or replicates a JEN bundle must formally cite *both* the DOI of the original paper *and* the DOI of the bundle itself. This practice aligns with the emerging best-practice principles for data and software citation advocated by groups like FORCE11.
- **Tracking and New Metrics:** This system of dual citation enables a new, richer form of impact assessment.
    - **Integration with Altmetrics:** JEN will partner with services like **Altmetric** and **Dimensions** to track the usage and mention of bundle DOIs across the web, including in social media, blogs, policy documents, and other scholarly articles. This provides a broad view of the artifact's reach.
    - **Developing "Executable Impact" Metrics:** More importantly, JEN will pioneer and prominently display a new class of metrics on each article page, moving beyond simple citation counts to measure the actual use and validation of the computational artifact:
        - **Re-execution Count:** A counter, powered by logs from the journal's execution sandbox, showing how many times readers have successfully run the bundle.
        - **Fork/Derivative Count:** A metric tracking how many new submissions to JEN are explicitly based on a "fork" of a previously published bundle.
        - **Verification Score:** A count of how many published "Replication Reports" have successfully validated or extended the findings of the original bundle.

This Executable Citation Index fundamentally alters the academic credit economy. It creates a new, quantifiable currency for reproducibility. A researcher can now demonstrate impact not only by the number of times their paper is cited, but by the number of times their model has been run, validated, and built upon by their peers. This provides a powerful, career-relevant incentive for researchers to invest the necessary effort in creating high-quality, robust, and truly reproducible computational work, directly weaponizing the academic reward system to drive the desired cultural change.

A Phased Strategy for Launch and Adoption

The launch of the *Journal of Executable Neuroscience* requires a deliberate, phased strategy designed to build credibility, cultivate a community of early adopters, and scale its impact over time. A new journal, particularly one with such a high technical and conceptual bar for entry, cannot succeed with a simple open call for papers. It must first establish its brand and prove its value proposition with high-quality, exemplar content. This section outlines a three-phase, multi-year roadmap for launching JEN and establishing it as the definitive standard for verifiable research in its field.

Phase 1 (Year 1): Pilot Issue & Precedent Setting

The primary goal of the first year is to demonstrate the feasibility and value of the entire JEN workflow, from submission to interactive publication. This phase is about curation and quality control to set a high standard from the outset. A common failure mode for new journals is an initial lack of high-quality submissions; an invitation-only pilot phase mitigates this risk by ensuring the inaugural issue is of exceptional scientific and technical quality, creating a powerful first impression.

- **Actions:**
    1. **Establish Governance:** Recruit the founding Board of Directors, the Executive Editorial Board (Editor-in-Chief and Associate Editors-in-Chief), and the Technical Advisory Board. The involvement of recognized key opinion leaders in computational neuroscience and open science is critical for immediate credibility.
    2. **Build Core Infrastructure (v1.0):** Develop and deploy the initial technical stack. This includes customizing the Open Journal Systems (OJS) portal, setting up the CI/CD pipeline using GitHub Actions, deploying a dedicated BinderHub for the execution sandbox, and establishing the automated archival workflow with Zenodo.
    3. **Solicit Flagship Articles:** The Editor-in-Chief will personally invite 3-4 leading research groups, known for their commitment to open science and high-quality computational work, to submit papers for the pilot issue. These "friendly" labs will work closely with the JEN technical team to refine the process of creating a Reproducibility Bundle.
    4. **Commission the First Replication Report:** To showcase the journal's full ecosystem, one of the invited labs will be commissioned to write the first "Replication Report." This report will take the bundle from another pilot article, re-execute it, and perform a simple robustness analysis. Publishing these two articles side-by-side will powerfully demonstrate the virtuous cycle of verification that JEN enables.
    5. **Launch and Promote:** Publish the pilot issue with extensive marketing and outreach. The narrative will focus on these flagship papers as concrete examples of the future of scientific publishing, highlighting the ease with which readers can interact with and verify the research.

Phase 2 (Years 2-3): Community Building & Broader Outreach

With the precedent set and the workflow proven, the second phase focuses on scaling the journal's operations, growing the submission base, and embedding JEN's standards within the broader research community.

- **Actions:**
    1. **Open for General Submissions:** Issue a broad, public call for papers. The initial call will be thematically framed around solving the reproducibility crisis in neuroscience, attracting early adopters from labs that are frustrated with the current fragmented and unreliable state of computational research.
    2. **Community Training and Education:** Launch a series of workshops and tutorials,

held at major international conferences like the Society for Neuroscience (SfN), Computational and Systems Neuroscience (Cosyne), and the Organization for Computational Neurosciences (CNS). These events will provide hands-on training for researchers on how to create a JEN-compliant Reproducibility Bundle.
3. **Educational Partnerships:** Develop partnerships with graduate programs and computational neuroscience summer schools to integrate the "JEN standard" into the training of the next generation of researchers. This will help make the creation of reproducible artifacts a default practice rather than an afterthought.
4. **Launch the Executable Citation Index:** Deploy the public-facing dashboard on the JEN website that displays the novel impact metrics (Re-execution Count, Fork/Derivative Count, Verification Score) for each published article. This makes the new reward system visible and tangible, further incentivizing high-quality submissions.

Phase 3 (Years 4+): Standardization and Policy Influence

In the final phase, JEN will leverage its established reputation and community to drive systemic change, influencing standards and policies across the scientific publishing and funding landscape. The long-term vision is not merely for JEN to be a successful niche journal, but for its core principles to become the expected standard everywhere.

- **Actions:**
    1. **Formalize the Bundle Standard:** Work with the Technical Advisory Board and community partners to formalize the "JEN Reproducibility Bundle" specification. This will involve publishing a detailed technical document and submitting the RO-Crate profile as a formal community standard, making it easier for other tools and platforms to adopt.
    2. **Influence Funder Mandates:** Use the journal's body of work and impact metrics as evidence in advocacy efforts directed at funding agencies (NSF, NIH, ERC). The goal is to persuade them to recommend or mandate the submission of JEN-compliant, executable bundles as a required deliverable for grants in computational fields. This would align funding incentives directly with the goal of operational reproducibility.
    3. **Scale the Model:** Explore pathways for other journals to adopt the JEN model. This could involve offering JEN's CI/CD pipeline and technical review process as a shared, non-profit service to other publishers who wish to implement an executable paper track.
    4. **Expand the "Executable" Brand:** Once the model is firmly established in neuroscience, consider expanding the scope by launching sister journals under a broader "Journal of Executable Science" umbrella, targeting fields like computational biology, bioinformatics, and machine learning, which face identical reproducibility challenges.

This phased strategy provides a clear and actionable path from a visionary concept to a field-defining institution. By starting with a curated, high-quality pilot, building a community of practice, and then leveraging that community's success to influence broader policy, JEN can achieve its ultimate mission: to make operational, verifiable reproducibility the new standard for computational science. The journal's greatest success will be realized when its own unique selling proposition becomes the baseline expectation for all scientific publishing.

Conclusion

The current paradigm of scientific publishing in computational fields is no longer sufficient. It is predicated on a model of trust that has been strained by a systemic failure to ensure verifiability. The evidence from large-scale replication projects is clear: the static, narrative paper is an inadequate vehicle for communicating complex, dynamic computational research. The result is a scientific ecosystem plagued by economic inefficiency, inflated claims, and a crisis of confidence.

The *Journal of Executable Neuroscience* represents a direct and comprehensive response to this challenge. It is not an incremental improvement but a fundamental rethinking of what a scientific publication can and should be. By shifting the core unit of publication from the manuscript alone to the manuscript plus a mandatory, self-contained, and computationally verified **Reproducibility Bundle**, JEN makes several transformative moves:

1. **It Realigns Responsibility:** It places the burden of ensuring reproducibility where it belongs—with the author at the time of submission—rather than leaving it as an unfunded externality for the community to solve. The automated CI/CD pipeline acts as an objective and efficient gatekeeper, enforcing this new standard without consuming precious reviewer and editor time.
2. **It Transforms the Reader Experience:** It converts the passive act of reading a paper into an active, interactive process of scientific exploration. The "Run Paper" button is not a gimmick; it is a tool for deeper understanding, for verification, and for building upon prior work with unprecedented ease.
3. **It Creates a Virtuous Ecosystem:** Through its carefully designed taxonomy of article types, particularly the inclusion of "Replication Reports," JEN builds a self-sustaining feedback loop where the verification and extension of prior work are valued, citable contributions to the scientific record.
4. **It Intervenes in the Academic Reward System:** The "Executable Citation Index" is the journal's most potent tool for cultural change. By creating a new class of metrics that track the use, reuse, and validation of the computational artifacts themselves, JEN provides a tangible, career-relevant incentive for researchers to invest in the hard work of producing robust, high-quality, and truly reproducible science.

The strategic plan laid out in this document provides a detailed roadmap for realizing this vision. It is ambitious but grounded in a realistic assessment of the existing technological and social landscape. It leverages successful precedents from innovative journals like *eLife* and *ReScience C*, integrates mature open-source technologies like BinderHub and RO-Crate, and proposes a governance model designed for long-term, community-led sustainability.

The launch of the *Journal of Executable Neuroscience* is more than the creation of a new academic venue. It is a declaration that for computational science to fulfill its promise, its methods and its results must be not only published but proven. It is a commitment to building the infrastructure, the standards, and the culture necessary to move from an era of archival availability to a new standard of operational, verifiable, and executable science.

Works cited

1. Editorial: Reproducible analysis in neuroscience - Frontiers, https://www.frontiersin.org/journals/neuroinformatics/articles/10.3389/fninf.2024.1520012/full 2. Reproducibility Project: Cancer Biology | Collections - eLife,

https://elifesciences.org/collections/9b1e83d1/reproducibility-project-cancer-biology 3. eLife, the Center for Open Science, and Science Exchange partner to assess the reproducibility of cancer biology research, https://www.bitss.org/cancer-studies-reproducibility/ 4. Reproducibility in Cancer Biology: What have we learned? - eLife, https://elifesciences.org/articles/75830 5. Reproducibility: How replicable is biomedical science in Brazil? | Interviews - eLife, https://elifesciences.org/interviews/388d5959/brazilian-reproducibility-initiative 6. Code Policy | Open science | Springer Nature, https://www.springernature.com/gp/open-science/code-policy 7. Materials, Software and Code Sharing | PLOS Computational Biology, https://journals.plos.org/ploscompbiol/s/materials-and-software-sharing 8. Code Availability | PLOS Computational Biology - Research journals, https://journals.plos.org/ploscompbiol/s/code-availability 9. Information for Authors | Journal of Neuroscience, https://www.jneurosci.org/content/information-authors 10. Reproducibility in Computational Neuroscience Models and Simulations - ResearchGate, https://www.researchgate.net/publication/297676571_Reproducibility_in_Computational_Neuroscience_Models_and_Simulations 11. Reproducibility in Computational Neuroscience Models and Simulations - PMC, https://pmc.ncbi.nlm.nih.gov/articles/PMC5016202/ 12. The five pillars of computational reproducibility: bioinformatics and beyond - Oxford Academic, https://academic.oup.com/bib/article/24/6/bbad375/7326135 13. Making an executable paper with the Python in Heliophysics Community to foster open science and improve reproducibility - Frontiers, https://www.frontiersin.org/journals/astronomy-and-space-sciences/articles/10.3389/fspas.2022.977781/full 14. Creating an executable paper is a journey through Open Science, https://csh.ac.at/publication/creating-an-executable-paper-is-a-journey-through-open-science/ 15. Opening the Publication Process with Executable Research Compendia - D-Lib Magazine, https://www.dlib.org/dlib/january17/nuest/01nuest.html 16. Data Publications & Reproducibility - Data Cooperative - The University of Arizona, https://data.library.arizona.edu/data-management/best-practices/data-publishing-reproducibility 17. Reproducible Environments - The Turing Way, https://book.the-turing-way.org/reproducible-research/renv 18. ReScience C - Wikipedia, https://en.wikipedia.org/wiki/ReScience_C 19. ReScience C, http://rescience.github.io/ 20. ReScience C: A Journal for Reproducible Replications in Computational Science | Request PDF - ResearchGate, https://www.researchgate.net/publication/334104287_ReScience_C_A_Journal_for_Reproducible_Replications_in_Computational_Science 21. Editorial: Reproducibility in neuroscience - Frontiers, https://www.frontiersin.org/journals/integrative-neuroscience/articles/10.3389/fnint.2023.1271818/full 22. Sustainable computational science: the ReScience initiative - PMC, https://pmc.ncbi.nlm.nih.gov/articles/PMC8530091/ 23. About Open Journal Systems | Online Learning, https://olj.onlinelearningconsortium.org/index.php/olj/about/aboutThisPublishingSystem 24. Janeway - Home, https://janeway.systems/ 25. Continuous Integration Tools: Top 7 Comparison - Atlassian, https://www.atlassian.com/continuous-delivery/continuous-integration/tools 26. What is Continuous Integration | Atlassian, https://www.atlassian.com/continuous-delivery/continuous-integration 27. Continuous integration - Wikipedia, https://en.wikipedia.org/wiki/Continuous_integration 28. What is continuous integration? - The Turing Way, https://book.the-turing-way.org/reproducible-research/ci/ci-options 29. Code Ocean Platform - EBSCO, https://www.ebsco.com/sites/g/files/nabnos191/files/acquiadam-assets/Code-Ocean-Platform-Guide.pdf 30. Using Codeocean for sharing reproducible research, https://rse.princeton.edu/2021/03/using-codeocean-for-sharing-reproducible-research/ 31. Code Ocean software overview | Code Ocean, https://codeocean.com/product 32. Code Ocean and Springer Nature expand partnership across the Nature Portfolio, https://group.springernature.com/gp/group/media/press-releases/code-ocean-expands-across-nature-portfolio/27688688 33. Extension of partnership with Code Ocean will help Springer Nature authors to better share their code and data, https://group.springernature.com/gp/group/media/press-releases/code-ocean-and-springer-nature-extend-partnership/23717000 34. (PDF) Promoting reproducibility with Code Ocean - ResearchGate, https://www.researchgate.net/publication/349470246_Promoting_reproducibility_with_Code_Ocean 35. Universities | Code Ocean, https://codeocean.com/solutions/universities 36. Code Ocean VPC - AWS Marketplace, https://aws.amazon.com/marketplace/pp/prodview-gcpgmz747khn2 37. Code Ocean Introduces New Offering for Academic Research Labs, https://codeocean.com/press-releases/code-ocean-introduces-new-offering-for-academic-research-labs 38. jupyterhub/binderhub: Run your code in the cloud, with technology so advanced, it feels like magic! - GitHub, https://github.com/jupyterhub/binderhub 39. README.md - jupyterhub/binderhub - GitHub, https://github.com/jupyterhub/binderhub/blob/master/README.md 40. The BinderHub Architecture — BinderHub documentation, https://binderhub.readthedocs.io/en/latest/overview.html 41. Introduction to BinderHub - The Turing Way, https://book.the-turing-way.org/reproducible-research/binderhub/binderhub-introduction 42. Changing the culture of data science - The Alan Turing Institute, https://www.turing.ac.uk/about-us/impact/changing-culture-data-science 43. Towards frictionless, portable, and sustainable reproducibility with Binder | 2i2c, https://2i2c.org/blog/2025/frictionless-reproducibility/ 44. Publishing computational research - a review of infrastructures for reproducible and transparent scholarly communication - PMC, https://pmc.ncbi.nlm.nih.gov/articles/PMC7359270/ 45. About — Whole Tale 1.2.0.0 documentation - Read the Docs, https://wholetale.readthedocs.io/en/stable/README.html 46. Whole Tale — Whole Tale 1.2.0.0 documentation, https://wholetale.readthedocs.io/ 47. Developing, Packaging and Sharing Reproducible Research ..., https://www.dataone.org/webinars/developing-packaging-and-sharing-reproducible-research-objects-whole-tale-approach/ 48. Implementing Computational Reproducibility in the Whole Tale Environment - Victoria Stodden, https://www.stodden.net/papers/p17-chard.pdf 49. Welcome to a new ERA of reproducible publishing | Labs - eLife, https://elifesciences.org/labs/dc5acbde/welcome-to-a-new-era-of-reproducible-publishing 50. Stencila - Open VSX Registry, https://open-vsx.org/extension/stencila/stencila 51. stencila/stencila: Documents with Scientific Intelligence - GitHub, https://github.com/stencila/stencila 52. Stencila, https://stencila.io/ 53. dtgeo / Metadata / CWL + RO-Crate - GitLab, https://gitlab.com/dtgeo/metadata/cwl-ro-crate 54. Research Object Crate (RO-Crate), https://www.researchobject.org/ro-crate/ 55. RO-Crate (Research Object Crate) - EOSC Association, https://eosc.eu/roadmap/ro-crate-research-object-crate/ 56. Proposing a New RO-Crate Profile for Enhanced Reproducibility in Computational Experiments Eleni Adamidi[1], Panagiotis Deligiann - Zenodo, https://zenodo.org/records/11200097/files/Performance_RO_Crate_Profile_WCRI2024.pdf?dow nload=1 57. RO-Crate profiles - Research Object, https://www.researchobject.org/ro-crate/profiles.html 58. Board of directors | About - eLife, https://elifesciences.org/about/people/directors 59. About eLife, https://elife.elifesciences.org/about 60. About the Journal of Open Source Software, https://joss.theoj.org/about 61. Journal of Open Source Software (JOSS): design and first-year review - PMC, https://pmc.ncbi.nlm.nih.gov/articles/PMC7340488/ 62. Transparency & best practice - DOAJ, https://doaj.org/apply/transparency/ 63. Good governance: Investigating models in scholarly communication and open research, https://investinopen.org/blog/good-governance-investigating-models-in-scholarly-communication-and-open-research/ 64. Brain Science - Allen Institute, https://alleninstitute.org/division/brain-science/ 65. Implementing DOIs for Research Data - D-Lib Magazine, https://www.dlib.org/dlib/may12/simons/05simons.html 66. DOIs for Research Software: Increasing Visibility, Connectivity, Citability - DataCite, https://datacite.org/event/webinar-dois-for-research-software-increasing-visibility-connectivity-citability/ 67. How to cite and describe software, https://www.software.ac.uk/publication/how-cite-and-describe-software 68. Software citation - Editor Resources - Taylor & Francis, https://editorresources.taylorandfrancis.com/welcome-to-tf/digital-publishing-excellence/software-citation/ 69. Journal Production Guidance for Software and Data Citations - PMC, https://pmc.ncbi.nlm.nih.gov/articles/PMC10522580/ 70. Altmetric: Reveal online attention to research, https://www.altmetric.com/ 71. Researcher data access program - Altmetric, https://www.altmetric.com/our-audience/researchers/research-access/ 72. www.altmetric.com, https://www.altmetric.com/#:~:text=Altmetric%20tracks%20and%20captures%20a,the%20research%20you%20care%20about. 73. Reproducible Research - Dime Wiki - World Bank, https://dimewiki.worldbank.org/Reproducible_Research 74. A Look at Altmetrics and Its Growing Significance to Research Libraries - Deep Blue Repositories, https://deepblue.lib.umich.edu/bitstream/handle/2027.42/99709/UMLibraryAltmetrics_090513.pdf?sequence=1&isAll 75. Faculty - McGovern Institute for Brain Research - MIT, https://mcgovern.mit.edu/researchers/ 76. Computational Neuroscience Archives - McGovern Institute for Brain Research - MIT, https://mcgovern.mit.edu/research-area/computational-neuroscience/ 77. The State of Computational Neuroscience - Simons Foundation, https://www.simonsfoundation.org/2018/01/02/the-state-of-computational-neuroscience/