

Methods Hub: A Community-Driven, Interactive Platform for Open-Source Computational Tools and Tutorials

Christina Viehmann ¹ , Johannes Kiesel ² , Arnim Bleier ¹ , Chung-hong Chan ¹ ,
Po-Chun Chang ² , Raniere Gaia Costa da Silva ¹ , Ahrabhi Kathirgamalingam ¹ ,
Taimoor Khan ² , Stephan Linzbach ² , Fakhri Momeni ² , Felix Münch ² , Ran Yu ² ,
Stefan Dietze ^{2,3} , and Claudia Wagner ^{1,4} 

¹ Department for Computational Social Science, GESIS – Leibniz Institute for the Social Sciences, Germany

² Department of Knowledge Technologies for the Social Sciences, GESIS – Leibniz Institute for the Social Sciences, Germany

³ Heinrich Heine University Düsseldorf, Germany

⁴ RWTH Aachen University, Germany

Correspondence: Christina Viehmann (christina.viehmann@gesis.org)

Submitted: 31 January 2026 **Accepted:** 23 July 2026 **Published:** 8 October 2026

Issue: This article is part of the issue “Open Research Infrastructures and Resources for Communication and Media Studies” edited by Silke Fürst (University of Zurich), Johannes Breuer (Center for Advanced Internet Studies / University of Duisburg-Essen), Erik Koenen (University of Bremen), Dimitri Prandner (Johannes Kepler University of Linz), Christian Schwarzenegger (University of Bremen), and Christian Strippel (Weizenbaum Institute), fully open access at <https://doi.org/10.17645/mac.i504>

Abstract

As computational methods become increasingly central to social science research, there is a growing demand for research software infrastructures that are sustainable, grounded in principles of open science, and embedded in the community. In response, we are introducing the Methods Hub as an open-source, community-driven collection of computational tools designed to support social science research in solving complex data-related tasks. The Methods Hub integrates three complementary components: First, quality-curated content, including computational methods and accompanying tutorials, which are specifically presented from a social science research perspective. Second, the methods and tutorials are hosted on a web portal enabling researchers to discover, apply, learn, and publish computational methods. Third, interactive, browser-based execution backends allow users to directly run, test, and adapt the code of methods and tutorials. These components are guided by a set of core principles: a long-term infrastructure perspective, reproducibility requirements, and a focus on community. The Methods Hub builds a long-term infrastructure that extends beyond the scope of single projects. Further, reproducibility functions as a quality assurance measure while advancing the practices of open science. Finally, the Methods Hub is built on public, community-based submissions by treating research software as citable academic output and providing visibility, accessibility, and formal recognition. We are also broadening the community by fostering learning and ease of access. Looking ahead, the Methods Hub is exploring the potential of AI as both

a research tool and to further strengthen the infrastructure, while maintaining transparency, reproducibility, and quality standards for computational social science research.

Keywords

computational methods; computational social science; interactive execution; open science; reproducibility

1. Introduction

Research questions driven primarily by the abundance of digital behavioral data have become commonplace in the social sciences (e.g., Puschmann & Pentzold, 2021). This trend has been particularly pronounced in media and communication studies, where digital communication has long been equated with continuous data streams (Hilbert et al., 2019). A similar development can be observed in the adoption of computational methods, such as web scraping or argument mining. These approaches are now firmly embedded in the communication scholar's toolkit. Against this backdrop, some academics have identified a "post-computational turn" in the social sciences (Hepp et al., 2021).

This consolidation is further reflected in growing institutionalization: Universities have established dedicated computational social science (CSS) labs; professional associations such as the International Communication Association have created specialized divisions to foster exchange and establish methodological standards; and journals devoted explicitly to computational methods, such as *Computational Communication Research* (van Atteveldt et al., 2019), have emerged. At this stage of consolidation, the focus has shifted toward creating sustainable infrastructures grounded in dedicated communities and toward promoting transparency and reproducibility as standard research practices (cf. Haim, 2021; Hepp et al., 2021; Niemann-Lenz et al., 2019; Strippel, 2021; van Atteveldt et al., 2019).

The Methods Hub (<https://methodshub.gesis.org>) has been created to address these needs: It is an institutionally funded, open-source, community-driven platform that lowers the barriers for social scientists to gain access to computational methods while offering visibility, recognition, and standardization to method developers. It takes a specialized social science perspective, presenting all resources from the viewpoint of researchers seeking to answer socially relevant questions using research designs that involve complex data-related tasks. While Methods Hub users are mostly social scientists, the platform also includes computer scientists addressing similar research questions. In fact, method contributors are expected to come from diverse disciplinary backgrounds.

In concrete terms, the Methods Hub provides a curated collection of computational tools and tutorials that support different stages of the research workflow, from data collection to analysis and visualization. For example, a user interested in utilizing hashtag dynamics on Mastodon to trace the activities of a political movement can use *rtoot* (Schoch & Chan, 2025) to collect data, apply *ELFEN* (Maurer, 2025) to extract entities such as hashtags, and then proceed with a topical analysis of the posts (Khan, 2025) or a network analysis between users (Schoch, 2025). These steps are accompanied by tutorials documenting workflows that make it easier to understand how tools can be applied in practice.

The platform is designed for different user groups. For researchers who want to apply computational methods, it provides structured access to tools and tutorials across the full research data workflow. Content is organized by task and differentiated by difficulty level, allowing users with varying levels of experience to find suitable entry points. In addition, browser-based interfaces enable users to try out selected tools without installing software locally. For contributors who develop research software (e.g., R or Python packages, or analysis scripts), the Methods Hub offers an infrastructure which allows them to share their work. Contributions are curated and evaluated to ensure they are understandable, reusable in social science contexts, and citable as research outputs. The platform is also useful for teaching. Instructors, who are key players in advocating good scientific practice among students of social science, can rely on the Methods Hub's stable, versatile environment for the long-term, reproducible dissemination of complex workflows without having to assemble materials from multiple sources.

Overall, the Methods Hub connects computational tools to social science research practices by aligning documentation, terminology, and use cases. Its central aim is to make computational methods more accessible and reusable by providing transparent, well-documented representations of tools, while treating reproducibility as a core requirement rather than an additional feature (cf. Bleier et al., 2026).

In Section 2 we describe the three main components of the Methods Hub: (a) we introduce the methods and tutorials content categories and elaborate on our content strategy and collection process; (b) we present the functionalities of the web portal for hosting methods and tutorials; and (c) we explain the interactive execution environments that enable users to directly run, test, and modify code in the browser. We then turn to the three core principles that guide the Methods Hub's strategy, elaborating on the vision that underlies its design and development: a long-term infrastructure, reproducibility requirements, and a focus on community. We conclude with an overview of ongoing goals for the infrastructure's development and an outlook on the future.

2. Components and Functionalities of the Methods Hub

The Methods Hub is organized into three major components: (a) content, (b) a web portal, and (c) interactive environments. The Methods Hub uses these components to support its major functionalities: discovering, reusing, learning, applying, and sharing computational methods. The Methods Hub prototype was launched internally in January 2025, following a three-year conception and development phase. After further testing and refinement, the Methods Hub was introduced to the public in October 2025.

2.1. Content

2.1.1. Content Categories and Curation

The centerpiece of the Methods Hub is its content: a curated collection of open-source resources designed to support researchers in collecting, preprocessing, analyzing, visualizing, and evaluating diverse types of social science data. Additionally, we provide content to support the setup of reproducible computational workflows. With this range of content, the Methods Hub aims to cover all data-related tasks in the empirical research process of (computational) social science. The Methods Hub addresses the needs of researchers with varying levels of expertise and research interests. To achieve this, the Methods Hub offers two complementary content formats: methods and tutorials.

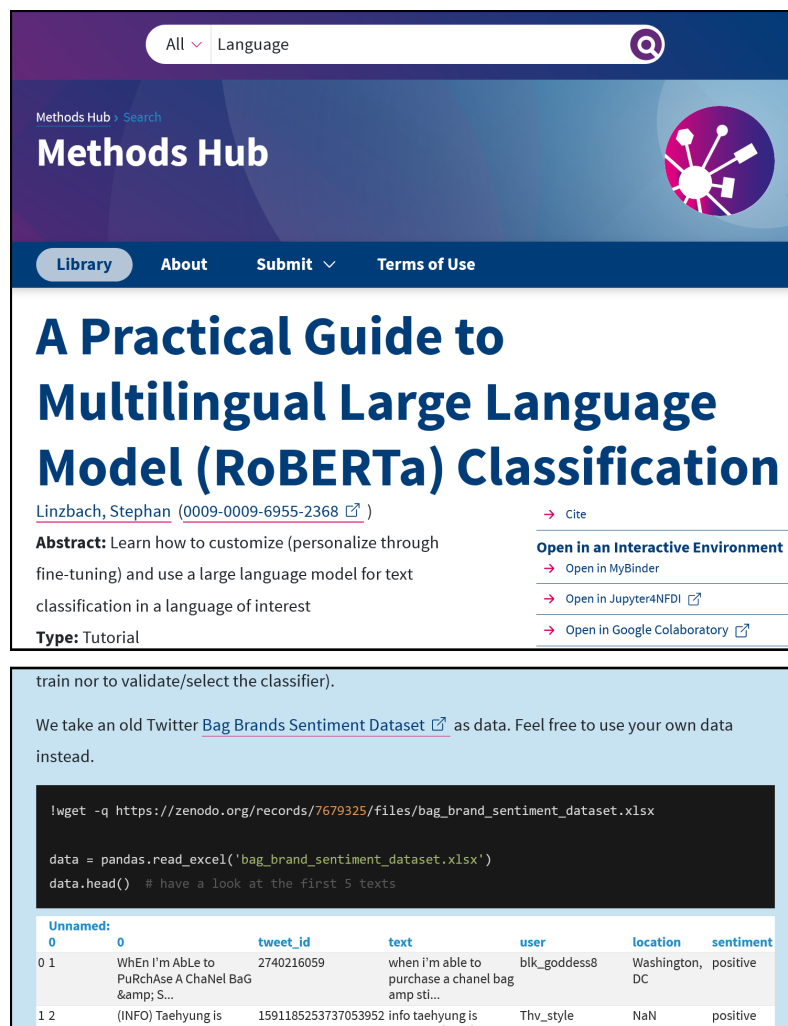
Methods in the Methods Hub refer to systematic computational procedures, such as code snippets or software packages, used in social science research. To adhere to the Methods Hub vision of openness, methods have to be implemented in an open-source programming language (such as Python or R), be publicly available on a code-sharing platform (such as GitHub), and provide an open license to reuse (such as an MIT license). The Methods Hub systematically showcases methods using a standardized presentation format. At a glance, users are able to find key information about each method, including the contributor, a short description, relevant topics and keywords, and the programming language used. In addition, each method page provides a detailed description, sample use cases, information on input and output data, the setup required, and important technical details. At their core, the pages feature a special section that explains how to use the method in practice. This separation of technical details and practical applications aims to make the content more accessible while retaining detailed documentation. The method pages for *sweater*, an R package for evaluating implicit biases in word embeddings (Chan, 2025), and the Telegram Data Collector, a Python-based method for extracting messages from Telegram channels (Gangopadhyay, 2025b), are among the many already available method pages and illustrate the Methods Hub's consistent framework.

Tutorials are instructional resources that support self-guided learning and focus on concrete research tasks relevant to social science research. They are explicitly tailored to the needs of social science research and combine methodological explanations with step-by-step code examples. While the methods showcase research software for social science research, the tutorials demonstrate how methods can be applied in practice. A tutorial can include and combine multiple methods. Thus, tutorials bridge the gap between methods and real-world research tasks and applications. Tutorials on the Methods Hub also have a standardized display to facilitate both navigation and contribution (cf. Figure 1). Users can quickly find information on the tutorial's difficulty, estimated duration, topics, keywords, and the programming language. Tutorial pages also state the learning objectives, target audience, use cases, and the required setup. At their core, tutorials provide step-by-step guidance organized into shorter chapters, while allowing contributors some flexibility to structure the content in a way that best supports learning. For example, currently available tutorials offer insights into conducting basic network analysis in R (Schoch, 2025) and detecting languages using Python (Gangopadhyay, 2025a).

Together, methods and tutorials provide a comprehensive, flexible learning environment. Users can either directly adopt tools for their own projects or follow guided examples to understand when and how to apply specific computational techniques. This dual structure supports a wide range of use cases, from quick issue resolution to methodological exploration.

The content on the Methods Hub is curated and quality controlled. These two steps, content acquisition and quality assurance, are supported using a variety of measures.

We have assembled a team of editors to curate content according to a set of quality criteria. These editors are selected based on their knowledge of computational tooling ecosystems (e.g., CRAN, PyPI), their own tool development and experience in tool documentation, and their experience with interdisciplinary collaboration in CSS research. Their most important task is to ensure that submissions are relevant to the social sciences and fall within the methodological scope of the Methods Hub, i.e., methods and tutorials should support the collection, processing, analysis, or visualization of data in social science research (see our taxonomy of content categories at <https://methodshub.gesis.org/submit/taxonomy>). This relevance and



The screenshot shows the Methods Hub interface. At the top, there's a search bar with 'All' and 'Language' filters. The main header 'Methods Hub' is prominent. Below it, a navigation bar includes 'Library', 'About', 'Submit', and 'Terms of Use'. The featured article is 'A Practical Guide to Multilingual Large Language Model (RoBERTa) Classification' by Linzbach, Stephan (0009-0009-6955-2368). The abstract describes learning to customize and use a large language model for text classification. The article type is 'Tutorial'. On the right, there are links to 'Cite', 'Open in an Interactive Environment' (MyBinder, Jupyter4NFDI, Google Colaboratory), and 'Open in MyBinder'. Below the article title, there's a code block showing how to download the 'Bag Brands Sentiment Dataset' and a preview of the dataset's first few rows.

```
!wget -q https://zenodo.org/records/7679325/files/bag_brand_sentiment_dataset.xlsx

data = pandas.read_excel('bag_brand_sentiment_dataset.xlsx')
data.head() # have a look at the first 5 texts
```

Unnamed: 0	0	tweet_id	text	user	location	sentiment
0 1	WhEn I'm AbLe to PuRchase A ChaNel BaG & S...	2740216059	when i'm able to purchase a chanel bag amp sti...	blk_goddess8	Washington, DC	positive
1 2	(INFO) Taehyung is ...	1591185253737053952	info taehyung is ...	Thv_style	NaN	positive

Figure 1. Detail page for a content item on the Methods Hub.

usefulness can be demonstrated either by the method's purpose or the use cases provided. In addition to their relevance to research questions within social sciences, methods and tutorials are assessed based on the quality of documentation and on computational reproducibility (see also Haim, 2021).

Content curation is similarly based on a targeted outreach strategy led by the Methods Hub editors as well as initiatives from contributors. Editorial outreach is informed by ongoing trend analyses and direct engagement with user groups via multiple channels: active participation in and organization of conferences, workshops, and demos (for an overview of our outreach activities, see <https://methodshub.gesis.org/community>); regular consultations with domain experts and internal evaluation groups; and continuous monitoring of current literature to identify emerging methodological trends.

In addition, the Methods Hub enables contributors to submit methods and tutorials directly via the web portal, providing an open community-based contribution pathway to complement the editorial acquisition process. To lower the barriers to submission, guidelines and templates are provided to assist with the submission process. Besides facilitating content preparation, these guidelines and templates are core measures used to ensure the quality of the content.

2.1.2. Submission Procedure

To be recognized as academic output, software implementations of computational methods and their tutorials are often treated as secondary artifacts of academic publications (Haim, 2021). The Methods Hub is contributing to the professionalization of research software development by accepting software implementations and educational materials as primary academic contributions.

Following examples such as the *Journal of Open Source Software* (Smith et al., 2018) and the criteria proposed by Haim (2021), we have derived a developer-friendly submission procedure for open-source content (cf. Figure 2). Submitters are expected to organize and version-control their content in a publicly accessible Git repository (e.g., GitHub). The main requirements for submission are a Git URL and commit hash from which we can “clone” the repository.

In these repositories, submitters license their code with an open-source license and include additional configuration files (see Section 2.3). An open license provides the legal basis for the collaborative and cumulative development of research software, enabling others to reuse and adapt the code to additional research contexts (Haim, 2021). The most important element that submitters are expected to include is a text file written using literate programming techniques, called the “Methods Hub-friendly README file.” Using literate programming systems such as Quarto or Jupyter notebooks, submitters can combine a description of the tool in English with code samples (referred to as “code blocks”) to demonstrate how the tool works. These examples are then executed and their outputs displayed. Next, this README file is converted to HTML and used as the landing page on the Methods Hub, serving as the primary and authoritative representation of the method or tutorial being submitted (cf. Figure 1).

To ensure a standardized display, we provide submitters with templates outlining the structure, making it easier to create complete and well-organized pages. The templates have been deliberately designed to accommodate a wide range of formats, including R or Python packages, scripted analysis workflows, and web applications with graphical user interfaces. As the Methods Hub grows, these templates will ensure the seamless integration of new content and will maintain quality.

During the actual submission, the link to the Git repository, along with a concise set of metadata, is collected via a submission form to ensure that all essential information is provided in a complete and standardized manner. After submission, the Methods Hub editors review the content using the published quality criteria mentioned above. Apart from technical requirements, such as whether the README file can be correctly converted into HTML, the most important criteria are relevancy to social science research, quality of documentation, and reproducibility. If the submission does not meet the criteria, contributors will be asked to revise and resubmit their content. If the submission satisfies these criteria, the submission will be published on the Methods Hub and a digital object identifier (DOI) will be issued (see Section 2.2).

In the long term, maintaining content is one of the central challenges when we attempt to keep research software usable. In our review, we ensure that dependency management via the standard files of the programming language in question is used in the content repositories, and we test its sufficiency using automated Docker container builds (see also Section 2.3). As long as the dependencies are available, this setup allows one to run the content. However, deprecated or obsolete dependencies may still lead to methods or tutorials that cannot be reproduced as intended.

Our basic assumption is that the submitters bear responsibility for maintenance. At the same time, we provide infrastructure that facilitates updates via versioned DOIs. In addition, we implement soft reminder mechanisms to identify and address issues (cf. Figure 2). First, a feedback channel directly linked to the original repository ensures that submitters are notified of problems, so that we do not rely on them to monitor the Methods Hub themselves. Using this channel, users can report issues such as deprecated application programming interfaces. Second, we are developing benchmark datasets that enable methods to be tested over time, allowing performance to be tracked and degradation detected. Third, we are exploring AI-based approaches to support automated reproducibility monitoring during regular editorial reviews.

If a method or tutorial no longer meets quality standards, it is archived: The resource remains accessible via its DOI for long-term reference, but it is no longer listed as active content in the library.

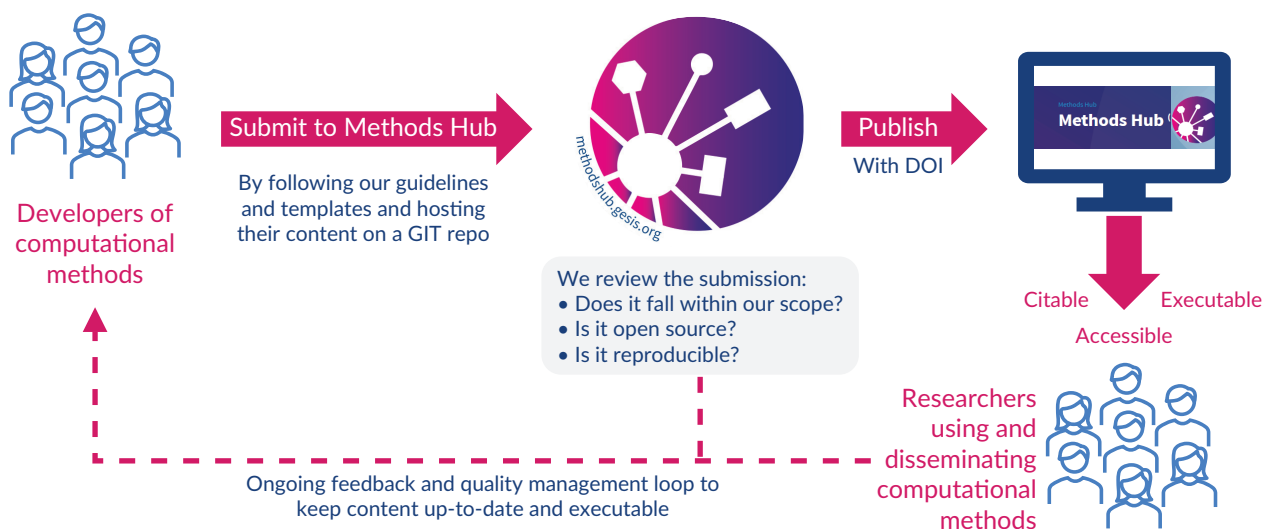


Figure 2. Methods Hub content submission lifecycle.

2.2. Web Portal

The Methods Hub is hosted on a web portal (<https://methodshub.gesis.org>) that has been developed to support a range of user-oriented functionalities, enabling social science users to search, access, download, execute, learn, cite, and publish computational tools. The platform is based on the Django framework and additional components such as Ansible, Docker, and Playwright.

The web portal serves as the entry point for finding methods and tutorials available on the Methods Hub: It consists of a curated library that enables users to efficiently search for computational methods relevant to their research needs (cf. Figure 3). Methods Hub's search function is provided by GESIS Search, an established research data discovery infrastructure (Hienert et al., 2019). A useful feature of the search function is the simultaneous discovery of related resources, such as datasets and publications. To enable effective searches and interoperability, all Methods Hub content items are mapped to categories of an established metadata schema (schema.org), the de facto standard for structured data on the web. Schema.org is supported by major search engine providers and has been widely adopted across websites, ensuring both visibility and interoperability.

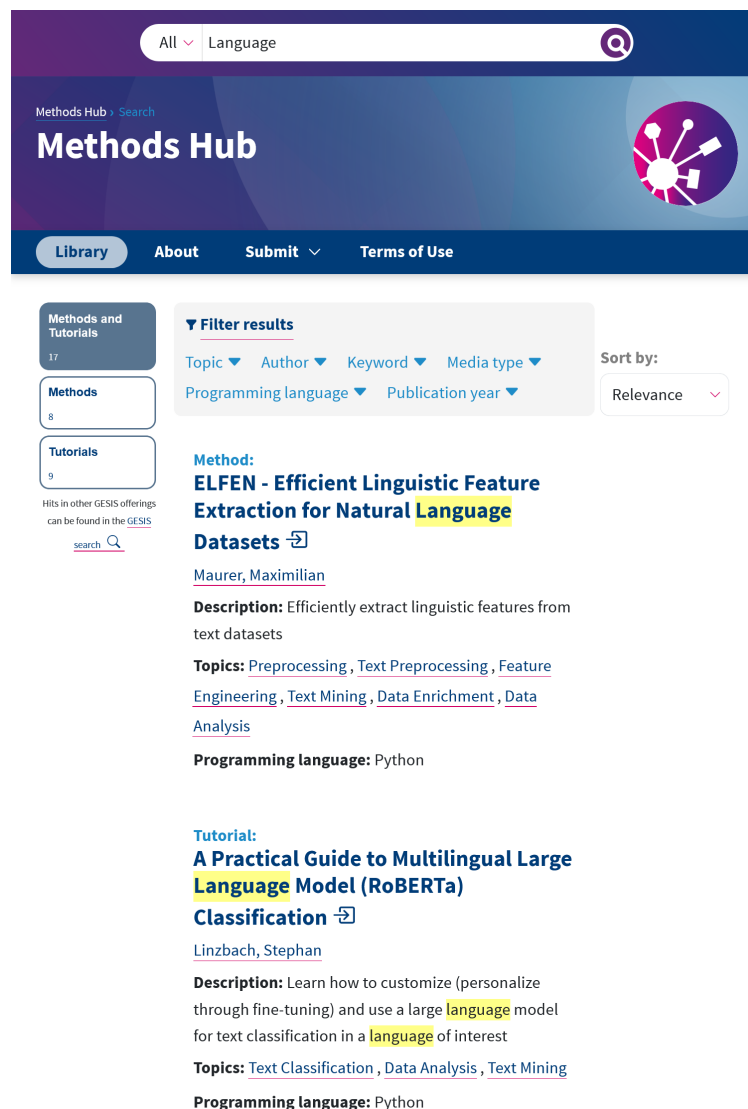


Figure 3. Library allowing users to browse through methods and tutorials on the Methods Hub.

Once users identify a method of interest, they can access structured information about it, including contributors, keywords, details about the hardware required, and its core functionalities (cf. details in Section 2.1.1). This standardized presentation of research software helps learners quickly find key information, compare resources, and understand how to apply the software in practice. To facilitate navigation and contextualization, each content item page features links to related items (ranging from methods to tutorials and academic articles) and displays key metadata that enable users to explore additional items within the same content category.

The search and access functionalities are closely aligned with transparency requirements central to the open science paradigm (Dienlin et al., 2021; Haim, 2021). They promote the reuse of computational methods through dedicated download and citation functionalities. To simplify reuse, users can download all relevant files associated with a method or tutorial via a single download link, including the source code, documentation, sample datasets, and configuration files specifying dependencies and system requirements.

The citability of Methods Hub content items is ensured by assigning DOIs via DataCite (<https://datacite.org>). For method users, DOIs guarantee long-term findability and persistent referencing. For method developers, they provide a formal recognition of software development as an academic contribution. All content items undergo an editorial review prior to publication; DOI assignment takes place following this evaluation and thus serves as a quality assurance measure (cf. Figure 2), confirming that the defined standards and requirements have been met. In this sense, the assignment of DOIs serves both as a mark of quality and as an anchor for academic recognition (Haim, 2021; Stecker et al., 2024).

Lowering barriers to the access and application of computational methods is one of the Methods Hub's central goals (see also Section 3.3), so learning is a conceptual core principle. In addition to methods, the platform hosts tutorials that enable users to familiarize themselves with new approaches (see content strategy in Section 2.1.1). To further enhance learning, the Methods Hub promotes direct code execution by providing interactive execution environments. These allow users to apply, test, and modify code from methods or tutorials directly within a browser-based environment (see Section 2.3).

Finally, in pursuing the vision of an informational infrastructure understood as a curated collection of research software, the Methods Hub relies on the voluntary contributions of researchers from the field (see Section 3.3). Beyond serving method users, the platform explicitly addresses researchers from the social sciences and computer science who wish to disseminate their software development work to a broader audience. By publishing methods or instructional resources, contributors make their work accessible to the wider scientific community, which in turn results in increased use of the tool in subsequent research (Stecker et al., 2024). In doing this, the Methods Hub fosters collaboration and knowledge sharing among researchers working in related domains while increasing the visibility and recognition of their contributions. At the same time, the Methods Hub recognizes software development as an ongoing process by implementing a versioning system that enables submitters to update methods or tutorials while preserving earlier versions using versioned DOIs.

2.3. Interactive Execution Environments

The third component that the Methods Hub provides consists of interactive execution environments. Because the content on the Methods Hub is implemented based on the principle of literate programming mentioned above, using technologies such as Quarto or Jupyter Notebook, these formats also allow direct interactive execution from within a web browser (cf. Figure 4). In these interactive execution environments, users can directly test and adapt methods in their research context without the hassle of configuring and installing software on their own computers. The interactive execution environments on the Methods Hub act like a sandbox: Users can directly plug in their own data or experiment with different settings to encourage trial-and-error learning.

However, the difficult part of implementing this kind of interactive execution environment is ensuring that content can be reproducibly executed within it. According to Schoch et al. (2024), a computational execution environment comprises at least four moving parts (operating system, system components, programming language, and software libraries with their versions), and any deviation from these components can break computational reproducibility.

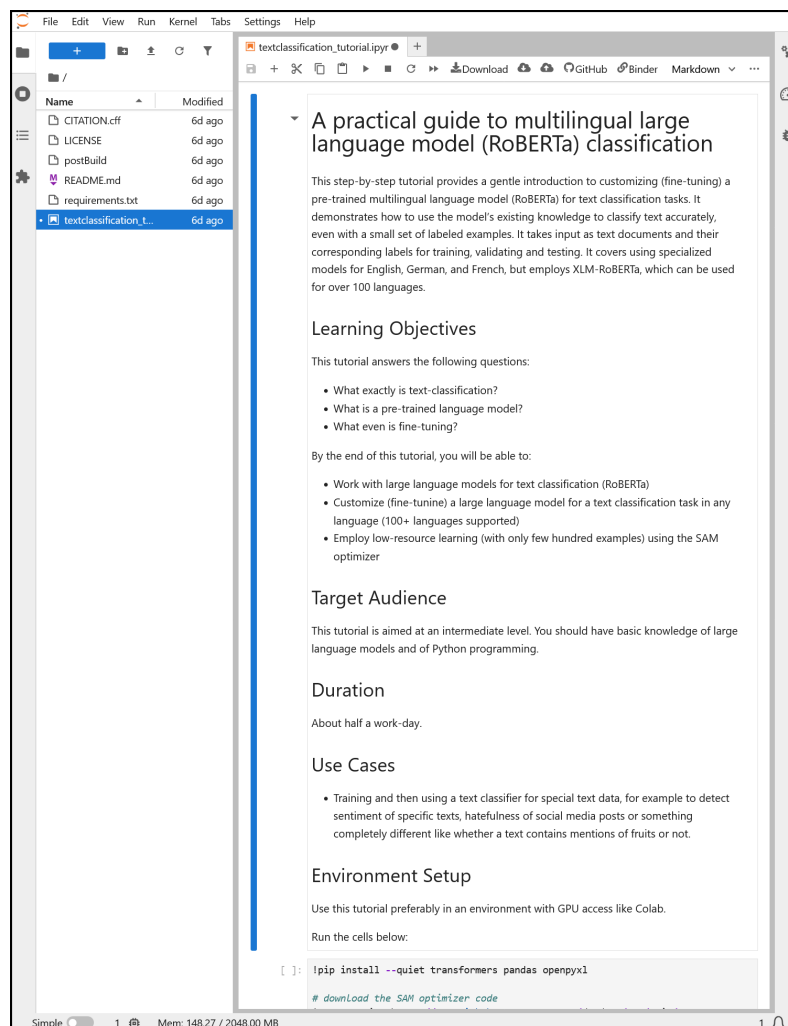


Figure 4. Content item view in the MyBinder interactive execution environment.

The content on the Methods Hub intended for interactive execution is expected to include repo2docker configuration files. Repo2docker (Forde et al., 2018) is the de facto open standard for specifying computational environments for computational research. It is able to cover all four moving parts and is easy to use: It makes use of commonly used methods for specifying software dependencies, and no significant effort or knowledge is needed to produce these configuration files. For instance, the execution environment is based on Docker. However, thanks to repo2docker, the submitter does not need to know Docker to configure such an environment. In the simplest case, the submitter only needs to provide requirements.txt (for Python) or install.R (for R, which contains calls to install.packages). Options to automatically generate these repo2docker configuration files are being explored with the aim of making future improvements.

By requiring configuration files that fully specify dependencies and execution contexts, the interactive execution backends can automatically construct reproducible computational environments for each content item. Once code is made executable through these environments, it is preserved in a form that enables long-term reproducibility in line with open science principles, independent of evolving software ecosystems. Because repo2docker is an open standard, the computational environment can be built on multiple interactive execution backends. Currently, the Methods Hub supports content execution on two

community-maintained interactive backends: MyBinder and Jupyter4NFDI. Both backends use the same underlying technology: non-commercial cloud services offering reproducible, interactive computing environments based on Docker containers and repo2docker. These interactive backends allow users to launch live computing sessions from content hosted in a Git repository. The only differences between the two options currently offered by the Methods Hub are eligibility and computing capacities. MyBinder (also used in other open science services, such as Harvard Dataverse) is free of charge but has limited computing capacity and memory. Also, the computing session ends when the browser is closed. It is more suited to trying out the content like a sandbox. Jupyter4NFDI, however, is primarily available to academics at universities and research institutes in Germany and therefore requires a login. A Jupyter4NFDI computing session is more persistent (e.g., it can be resumed) and provides greater computational power. It is suited to a more serious adoption and investigation of the methods and tutorials using one's own data.

The introduction of repo2docker-based interactive environments on the Methods Hub not only benefits users; it also provides a good opportunity to sensitize method developers to open standards and community-driven efforts that foster computational reproducibility. In doing so, it encourages a move away from reliance on proprietary solutions such as Google Colab or Amazon SageMaker.

3. Conceptual and Management Principles: A Long-Term, Community-Centered Infrastructure for Promoting Reproducible Computational Tools

As the CSS field moves from the emergent, exploratory phase into one of consolidation, the infrastructural choices made now will have lasting consequences for how methods are developed, shared, and applied. The Methods Hub has been conceptualized to address three interrelated, interdisciplinary needs that characterize the current stage of consolidation in CSS: (a) the establishment of sustainable research infrastructures, and (b) the advancement of reusability as a central principle of research software. Moreover, (c) it features community-building activities that cater to both perspectives, strengthen the currently fragmented community, and extend it by lowering the barriers to accessing, understanding, and applying computational methods across the social sciences. These needs point towards a broader goal: increased autonomy for the CSS community via infrastructure created and shaped by the people who use it.

These needs translate into the following core conceptual and management principles that underpin the Methods Hub: a long-term perspective, computational reproducibility as a conceptual premise, and a dedicated community focus that strives both to extend and deepen its community base. Translating these needs into explicit guiding principles is not merely a formality; it is a structural necessity, particularly given the audience's diversity. Without shared standards and conceptual commitments, a platform of this kind risks perpetuating the very fragmentation and inconsistencies in quality that it aims to overcome. Explicit principles also provide decision-making criteria when trade-offs arise, ensuring that the platform's direction remains stable amid personnel changes. By anchoring the Methods Hub in these commitments, we are positioning it as an active participant in shaping CSS norms rather than as a neutral repository, creating clear accountability criteria that the community can expect us to adhere to, and establishing the legitimacy that community-centered infrastructure depends on.

We now explain how each of these principles has been translated from a conceptual commitment to the specific design decisions and functionalities of the Methods Hub.

3.1. A Long-Term Infrastructure

First, the Methods Hub is designed as a long-term infrastructure for social science research software. Research software—understood as any “subset of computer code used by researchers to perform their research” (Yehudi et al., 2025, p. 2)—has become indispensable in data-intensive fields such as media and communication studies. However, infrastructures for developing, sharing, and reusing such software remain fragmented (Heft et al., 2023; Strippel, 2021; Wissenschaftsrat, 2011), with tools being largely confined to project-specific contexts and individual workflows. As a result, they are rarely designed for purposes of maintenance, reuse, or cumulative knowledge building (Haim, 2021; Hepp et al., 2021, 2023; Strippel et al., 2023).

Collections such as the Methods Hub function as informational infrastructures that organize and stabilize knowledge and tools for a research community (Strippel, 2021). To be effective, they require stable funding, institutional embedding, and dedicated staff (Wissenschaftsrat, 2011). The Methods Hub meets these requirements: it is institutionally embedded at GESIS, an infrastructure institute for the social sciences; and further integrated into the National Research Data Infrastructures, including its Data Science and AI consortium. It is run by a dedicated editorial and technical team and supported by stable funding, which is complemented by targeted third-party projects to drive innovation.

Further, Methods Hub supports sustainability by offering transparent documentation and the long-term availability of content items. All submissions undergo an editorial review to ensure that they are complete, clear, and functional, paving the way for long-term usability and recognition as academic output that contributes to sustainable research software ecosystems in the social sciences (Stecker et al., 2024). Long-term findability and reuse are supported by mapping the variety of research software artifacts to standardized metadata, issuing persistent identifiers, and implementing versioning that documents software development and preserves specific releases, a crucial aspect of reusability (see Section 3.2).

3.2. Reproducibility: The Methods Hub as a Showcase of Open Science

Moreover, the Methods Hub advances the principles of open science by emphasizing the roles of open-source computational methods and research software in enforcing research transparency and computational reproducibility. Against the backdrop of a widely diagnosed replication crisis in the social sciences (Dienlin et al., 2021; Open Science Collaboration, 2015), open science is regarded as a key corrective framework (Haim, 2021; Schoch et al., 2024; Strippel, 2021; Strippel et al., 2023). In the context of research software, open science translates into specific quality requirements, including transparency and reproducibility (Haim, 2021; van Atteveldt et al., 2019).

The Methods Hub is explicitly grounded in these requirements and operationalizes them with its infrastructural and procedural design choices. Regarding transparency, we have adopted the definition of transparency as “the process of making the content and process of producing evidence and claims transparent and accessible to others” (Munafò et al., 2017, p. 5). This extends beyond the mere publication of data and code: In practice, transparency also requires actionable access via findability. Without such findability, transparency does not support scrutiny and reuse (Dienlin et al., 2021; Haim, 2021). The Methods Hub addresses this by ensuring the long-term availability and discoverability of research

software using DOI assignments and standardized metadata. Additionally, standardized templates further promote the comprehensive and consistent documentation of methods and tutorials, thereby facilitating reuse, adaptation, and cumulative development.

In the context of CSS, reproducibility refers to the ability to rerun and inspect computational procedures so that others can reproduce results (Schoch et al., 2024). This reproducibility of tools and workflows is a necessary precondition for replicating findings and assessing their generalizability across datasets and research designs (Dienlin et al., 2021; Haim, 2021). This is why the Methods Hub requires research software submitted to us to be documented so that others can execute, inspect, and reproduce its functionality without relying on implicit assumptions, undocumented preparatory steps, or idiosyncratic computing environments. Accordingly, research software must be understandable, modular, and self-contained. This includes explicit documentation of analytical pipelines, parameters, dependencies, and versioning, along with clearly specified execution steps (Haim, 2021). These criteria are reflected in the submission guidelines (i.e., the literate programming paradigm) and are reinforced by editorial review processes that assess clarity, structure, and functionality.

Altogether, the Methods Hub is not merely a venue for open dissemination; it is an infrastructure that enables and stabilizes reproducible research practices. By embedding reproducibility into submission requirements, evaluation processes, and interactive execution environments, it serves as an infrastructural anchor for open science in social science research software.

3.3. Community-Centered: Broadening and Deepening CSS

Additionally, the Methods Hub is designed as a community-centered infrastructure that both broadens and deepens participation in CSS. It lowers barriers to entry while simultaneously fostering sustained collaboration and collective ownership of methodological development.

Broadening the community entails making computational methods accessible to a wider range of researchers across varying levels of technical expertise. Despite their growing importance, computational approaches remain unevenly distributed within the social sciences, with substantial structural, technical, and epistemic barriers limiting their adoption (Kathirgamalingam et al., 2025; Niemann-Lenz et al., 2019; Waldherr et al., 2025). Existing repositories and sharing platforms for computational tools and models either offer no reproducibility at all, or reproducibility that is questionable at best (e.g., commercial platforms with machine learning applications); they are typically oriented toward expert users and presuppose familiarity with computer-science-centric terminologies and practices (e.g., Hugging Face for pre-trained models), and offer little to no curation for the needs of social science (e.g., GitHub). In practice, this means that users often face an unstructured landscape where it is unclear what is relevant or where they should begin. The Methods Hub addresses these challenges by making research methods findable, understandable, and directly usable by different research communities. It assumes a translational role between computer science and the social sciences: Methods Hub explicitly frames the methodological relevance of computational methods in terms of social science research questions. Moreover, during the review of content items, a focus is on terminology, using commonly used phrases to compensate for the absence of a shared interdisciplinary knowledge base. Extensive documentation, commented code, and tutorials covering different levels of expertise support comprehensibility as a central quality criterion defining research software (Haim, 2021; Hepp et al., 2021). This educational dimension is further strengthened by interactive, browser-based

execution environments that enable hands-on experimentation and immediate application. By integrating these aspects, the Methods Hub helps to lower epistemic and technical barriers that often discourage social scientists from engaging with computational methods.

At the same time, the Methods Hub deepens the community by embedding contribution, collaboration, and long-term maintenance within the infrastructure itself: following a “from the community, for the community” principle, it treats researchers not only as users but as contributors to a shared and evolving methodological repertoire. The community collectively maintains, refines, and advances a shared methodology within the Methods Hub’s supportive structure. The infrastructure supports this collective effort by relieving contributors of infrastructural burdens while enabling them to transform project-based outputs into sustainable, citable resources. Using open calls, shared tasks, conference presentations, and workshops, the Methods Hub is continuously expanding and engaging its contributor base while incorporating feedback for ongoing refinement (cf. the Methods Hub community page, <https://methodshub.gesis.org/community>).

This community integration extends beyond individual actors to other infrastructures. The Methods Hub enables the sustainable integration of project-based initiatives, allowing their outputs to persist beyond funding cycles. In this way, the Methods Hub not only aggregates resources; it also stabilizes and connects fragmented infrastructural efforts within the field. One illustrative example is the KODAS Data Quality Toolbox, an educational infrastructure that supports researchers in developing competencies for data quality assessment. This toolbox was developed within the Competence Center Data Quality in the Social Sciences (KODAS) and was funded by the German Federal Ministry of Research, Technology, and Space from 2023 to 2026. To ensure the long-term preservation of the KODAS toolbox beyond the funding period, the Methods Hub will not only provide a sustainable home for individual content items; it will also support the maintenance of a recognizable, coherent data quality infrastructure.

By simultaneously broadening access and deepening collaboration, the Methods Hub is contributing to a more inclusive, interconnected, and sustainable CSS community (Lukito, 2024).

Taken together, these three principles position the Methods Hub as a sustainable infrastructure that not only supports CSS research, but which is also actively shaping the community and its standards, practices, and accessibility.

4. Outlook and Future Developments

The Methods Hub is a community-driven platform for sharing computational methods for analyzing digital behavioral data in the social sciences. This is particularly relevant for media and communication research, in which the study of digital media environments increasingly relies on computational approaches.

In sum, the Methods Hub addresses three cross-disciplinary needs: (a) it provides a sustainable infrastructure with a long-term perspective; (b) it advances the principles of open science by treating transparency and computational reproducibility as conceptual premises of software documentation; and (c) it serves as a community platform that both integrates research software development across the field and supports the dissemination of knowledge, skills, and practical experience across broader segments of the social sciences, contributing to a greater uptake of computational approaches.

To realize these goals, the Methods Hub is structured around three interrelated components. At its core is curated research software, referred to as methods, as the primary content. To support learning and broaden the application of computational methods, the Methods Hub includes tutorials as a second content category. Another component consists of interactive execution environments which enable users to directly run, test, and modify code from research software and tutorials in a browser-based environment, providing hands-on experience. The preparation and presentation of all content—particularly in relation to the interactive environments—are guided by the principles of reproducibility, with the aim of supporting computational reproducibility in CSS (Schoch et al., 2024). The technical backbone and gateway to these components is the web portal itself, which serves as the central point for discovering, browsing, and accessing content by means of a library and standardized metadata. Finally, by issuing persistent identifiers, the Methods Hub not only offers a venue for publishing research software; it also ensures citability and thus supports the academic recognition of software development.

In the years to come, the Methods Hub aims to establish itself as an infrastructure that sets national and international standards and provides best practices for the sharing, archiving, and reuse of computational methods in the social sciences. Achieving this goal means navigating a set of interrelated challenges—especially regarding quality assurance, increasing AI usage, software complexity, and software lifecycles—that will shape the Methods Hub's future development.

The first challenge concerns quality assurance. Ensuring high-quality content on the Methods Hub requires rigorous and constructive review processes. Our editorial workflows must support developers in improving their submissions without imposing disproportionate burdens, while also retaining our right to reject contributions that lack usefulness, reusability, methodological clarity, or ethical guardrails. Given the growing pressure on academic peer-review services, quality assurance cannot rely solely on peer review. This is why the Methods Hub is developing complementary pipelines for the automated testing of core functionalities and for assisting submitters in preparing materials that align with the needs, knowledge bases, and expectations of method and tutorial users—regarding aspects such as documentation, structure, and language. In this context, AI-based support tools hold considerable potential to upscale our content preparation, submission, and publication processes. We plan to explore these pathways in the near future; for example, we will be assessing AI assistance in the evaluation of content items that are submitted to us, using criteria such as relevance to the social sciences, clarity of expression, and the absence of formal errors. AI assistance may also be useful in the continuous monitoring and restoration of the executability of published content. Realizing this potential is not without challenges, however. To uphold the principles of open science, the Methods Hub stresses the importance of open-source and open-access LLMs to ensure the transparency, accountability, data privacy, and interpretability of model outputs with regard to assistant tasks such as the preparation, structuring, and reviewing content submissions (see also Hussain et al., 2024).

High-quality content also refers to how well computational methods perform on their intended tasks, as measured by benchmark datasets. This is why the Methods Hub aims to integrate an algorithmic benchmarking process. Integrating benchmark results into the Methods Hub will enable researchers to make informed decisions about which methods to employ for their specific task; for example, they may consciously choose a method that performs worse according to the benchmarks but which offers greater methodological clarity. Moreover, integrating benchmarking will enable researchers to see at a glance, years later, how much more effective newer methods are than those used previously, providing them with information as to whether to switch to newer methods for their current research.

Another challenge concerns the rapid diffusion of AI, both as an object of research and a methodological tool in CSS, raising more fundamental questions about the future role of method collections. The Methods Hub has actively engaged with this AI trend in several ways. It is positioning itself as an infrastructure facilitating AI-assisted social science research by providing instructional resources and guidance on responsible, transparent use. The Methods Hub is investigating the integration of computational methods that rely on large language models. The reproducibility of research based on these models, particularly closed ones, is especially low if they are applied without attention to provenance. The Methods Hub aims to address this provenance for the researcher by recording all the relevant parameters, thereby ensuring the reproducibility of research outcomes. Further, it deliberately aims to balance new AI-based approaches with more classical computational methods that cannot be easily replaced by generative models and which remain central to many research designs. Increased computational demand associated with AI-driven methods poses additional challenges, which the Methods Hub is addressing by partnering with complementary computational infrastructures, such as Jupyter4NFDI.

A further challenge concerns reusability in increasingly complex software ecosystems. Contemporary research software often relies on dense webs of interdependencies, many of which are external and subject to independent development cycles. To address this, the Methods Hub not only provides interactive execution environments based on container technologies; it is also contributing to the underlying infrastructure by actively participating in community-driven initiatives, such as the MyBinder federation. With this involvement, the Methods Hub is not merely positioning itself as a user of existing technologies, but as a contributor to innovation at the intersection of interactive execution and long-term reproducibility. This includes the ongoing evaluation of additional interactive environments that may complement the existing portfolio.

Nevertheless, besides the preservation of executable snapshots of current software states, there remain broader challenges related to software lifecycles (Yehudi et al., 2025). In particular, the long-term maintenance and continued development of open-source research software, which is necessary to keep pace with evolving research requirements and technological change, cannot be addressed by individual projects alone (Hepp et al., 2023). Strengthening these processes ultimately requires coordinated, community-level efforts across disciplines. In this landscape, infrastructures such as the Methods Hub can serve as key enablers by translating shared principles into sustainable practice. By foregrounding accessibility, reproducibility, and collective stewardship in CSS, the Methods Hub is well positioned to engage with the growing complexity of software and the integration of AI—both as part of the infrastructure and as an object of research—while maintaining high standards through transparent evaluation processes and benchmarking. As such, it offers a concrete pathway towards a more robust, reflexive, and future-oriented methodological ecosystem.

As a long-term infrastructure, the Methods Hub is not designed to resolve these challenges once and for all; the aim is to grow with them and with the community. This means we will be continuously adapting our processes, partnerships, and standards to remain a reliable foundation for CSS as the field evolves.

Acknowledgments

We gratefully acknowledge the valuable contributions of former Methods Hub team members David Schoch, Hajira Jabeen, Brigitte Mathiak, Gabriella Lapesa, Waleed Butt, Thomas Krämer, and Danilo Dessi to the Methods Hub project.

Funding

The work presented in this paper is financially supported by NFDI4DS (DFG project no. 460234259) and Jupyter4NFDI as part of Base4NFDI (DFG project no. 521453681). The publication of this article in open access was made possible by the institutional membership agreement between the GESIS – Leibniz Institute for the Social Sciences and Cogitatio Press.

Conflict of Interests

The authors declare no conflict of interests.

LLMs Disclosure

GPT-oss:120b and Overleaf Writefull have been used as language editing aids.

References

- Bleier, A., Kiesel, J., Viehmann, C., Münch, F. V., Chan, C.-h., Costa da Silva, R. G., Kathirgamalingam, A., Chang, P.-C., Khan, M. T., Linzbach, S., Momeni, F., Yu, R., Dietze, S., & Wagner, C. (2026). Improving reproducibility in computational social science with the Methods Hub. *Proceedings of the International AAAI Conference on Web and Social Media*, 20(1), 3038–3041. <https://doi.org/10.1609/icwsm.v20i1.42804>
- Chan, C.-h. (2025). *sweater* [Computer software]. GESIS Methods Hub. <https://doi.org/10.71627/sweater.1>
- Dienlin, T., Johannes, N., Bowman, N. D., Masur, P. K., Engesser, S., Kümpel, A. S., Lukito, J., Bier, L. M., Zhang, R., Johnson, B. K., Huskey, R., Schneider, F. M., Breuer, J., Parry, D. A., Vermeulen, I., Fisher, J. T., Banks, J., Weber, R., Ellis, D. A., . . . de Vreese, C. (2021). An agenda for open science in communication. *Journal of Communication*, 71(1), 1–26. <https://doi.org/10.1093/joc/jqz052>
- Forde, J., Head, T., Holdgraf, C., Panda, Y., Nalvarete, G., Ragan-Kelley, B., & Sundell, E. (2018, July 14). *Reproducible research environments with repo2docker* [Poster]. ICML 2018: Reproducibility in Machine Learning Workshop, Stockholm, Sweden. <https://openreview.net/forum?id=B1IYOwuoxm>
- Gangopadhyay, S. (2025a). *Detecting languages with Python: A step-by-step guide* [Tutorial]. GESIS Methods Hub. https://doi.org/10.71627/Language_Detection_Tutorial.1
- Gangopadhyay, S. (2025b). *Telegram data collector* [Computer software]. GESIS Methods Hub. <https://doi.org/10.71627/Telegram-Data-Collection.1>
- Haim, M. (2021). Gütekriterien und Handlungsempfehlungen für die Entwicklung von Forschungssoftware in der Kommunikations- und Medienwissenschaft. *M&K Medien & Kommunikationswissenschaft*, 69(1), 65–79. <https://doi.org/10.5771/1615-634x-2021-1-65>
- Heft, A., Jünger, J., Niemann-Lenz, J., & Possler, D. (2023). Die Nationale Forschungsdateninfrastruktur – eine Lösung infrastruktureller Bedarfe für die Inhaltsanalyse? *Publizistik*, 68(2), 355–376. <https://doi.org/10.1007/s11616-023-00789-5>
- Hepp, A., Hohmann, F., & Belli, A. (2023). Lasst uns unsere tools gemeinsam bauen! *Publizistik*, 68(2), 177–200. <https://doi.org/10.1007/s11616-023-00803-w>
- Hepp, A., Loosen, W., & Hasebrink, U. (2021). Jenseits des Computational Turn: Methodenentwicklung und Forschungssoftware in der Kommunikations- und Medienwissenschaft – zur Einführung in das Themenheft. *M&K Medien & Kommunikationswissenschaft*, 69(1), 3–24. <https://doi.org/10.5771/1615-634x-2021-1-3-1>
- Hienert, D., Kern, D., Boland, K., Zapilko, B., & Mutschke, P. (2019). A digital library for research data and related information in the social sciences. In *Proceedings of the 2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL)* (pp. 148–157). IEEE. <https://doi.org/10.1109/JCDL.2019.00030>

- Hilbert, M., Barnett, G., Blumenstock, J., Contractor, N., Diesner, J., Frey, S., Gonzalez-Bailon, S., Lamberso, P., Pan, J., Peng, T.-Q., Shen, C., Smaldino, P. E., van Atteveldt, W., Waldherr, A., Zhang, J., & Zhu, J. J. H. (2019). Computational communication science: A methodological catalyzer for a maturing discipline. *International Journal of Communication*, 13, 3912–3934. <https://ijoc.org/index.php/ijoc/article/view/10675/2764>
- Hussain, Z., Binz, M., Mata, R., & Wulff, D. U. (2024). A tutorial on open-source large language models for behavioral science. *Behavior Research Methods*, 56(8), 8214–8237. <https://doi.org/10.3758/s13428-024-02455-8>
- Kathirgamingam, A., Kulichkina, A., Bernhard-Harrer, J., & Hase, V. (2025). *Reflecting on social bias: Challenges and opportunities for computational social science*. OSF. https://doi.org/10.31235/osf.io/xr45y_v1
- Khan, T. (2025). *Latent dirichlet allocation* [Computer software]. GESIS Methods Hub. https://doi.org/10.71627/latent_dirichlet_allocation.1
- Lukito, J. (2024). Scholarly solidarity: Building an inclusive field for junior and minority researchers. *Political Communication*, 41(1), 152–161. <https://doi.org/10.1080/10584609.2023.2261876>
- Maurer, M. (2025). *ELFEN—Efficient linguistic feature extraction for natural language datasets* [Computer software]. GESIS Methods Hub. <https://doi.org/10.71627/elfen.1>
- Munafò, M. R., Nosek, B. A., Bishop, D. V. M., Button, K. S., Chambers, C. D., Percie du Sert, N., Simonsohn, U., Wagenmakers, E.-J., Ware, J. J., & Ioannidis, J. P. A. (2017). A manifesto for reproducible science. *Nature Human Behaviour*, 1, Article 0021. <https://doi.org/10.1038/s41562-016-0021>
- Niemann-Lenz, J., Bruns, S., Hefner, D., Knop-Huelss, K., Possler, D., Reich, S., Reinecke, L., Scheper, J., & Klimmt, C. (2019). Crafting a strategic roadmap for computational methods in communication science: Learnings from the CCS 2018 conference in Hanover. *International Journal of Communication*, 13, 3885–3893. <https://shorturl.at/KDGwZ>
- Open Science Collaboration. (2015). Estimating the reproducibility of psychological science. *Science*, 349(6251), Article aac4716. <https://doi.org/10.1126/science.aac4716>
- Puschmann, C., & Pentzold, C. (2021). A field comes of age: Tracking research on the internet within communication studies, 1994 to 2018. *Internet Histories*, 5(2), 135–153. <https://doi.org/10.1080/24701475.2020.1749805>
- Schoch, D. (2025). *Basic network analysis in R* [Tutorial]. GESIS Methods Hub. <https://doi.org/10.71627/basic-network-analysis-in-r.1>
- Schoch, D., & Chan, C.-h. (2025). *rtoot* [Computer software]. GESIS Methods Hub. <https://doi.org/10.71627/rtoot.1>
- Schoch, D., Chan, C.-h., Wagner, C., & Bleier, A. (2024). Computational reproducibility in computational social science. *EPJ Data Science*, 13(1), Article 75. <https://doi.org/10.1140/epjds/s13688-024-00514-w>
- Smith, A. M., Niemeyer, K. E., Katz, D. S., Barba, L. A., Githinji, G., Gymrek, M., Huff, K. D., Madan, C. R., Cabunoc Mayes, A., Moerman, K. M., Prins, P., Ram, K., Rokem, A., Teal, T. K., Valls Guimera, R., & Vanderplas, J. T. (2018). Journal of Open Source Software (JOSS): Design and first-year review. *PeerJ Computer Science*, 4, Article e147. <https://doi.org/10.7717/peerj-cs.147>
- Stecker, M., Balluff, P., Lind, F., Dinhopf, C., Waldherr, A., & Boomgaarden, H. G. (2024). Tools of the trade—When are software tools mentioned in computational text analysis research? *Computational Communication Research*, 6(1), 1. <https://doi.org/10.5117/CCR2024.1.6.STEC>
- Strippel, C. (2021). Forschungsinfrastrukturen für die Kommunikations- und Medienforschung im deutschsprachigen Raum. Initiativen, Bedarfe und Perspektiven. *M&K Medien & Kommunikationswissenschaft*, 69(1), 136–157. <https://doi.org/10.5771/1615-634x-2021-1-136>
- Strippel, C., Breuer, J., Fürst, S., Koenen, E., Prandner, D., & Schwarzenegger, C. (2023). Editorial: Data, archives,

and tools: Introducing new publication formats on infrastructures and resources for communication and media research. *Publizistik*, 68(2), 167–175. <https://doi.org/10.1007/s11616-023-00806-7>

van Atteveldt, W., Margolin, D., Shen, C., Trilling, D., & Weber, R. (2019). A roadmap for computational communication research. *Computational Communication Research*, 1(1), 1–11. <https://doi.org/10.5117/ccr2019.1.001.vana>

Waldherr, A., Weber, M., Wu, S., Haim, M., van der Velden, M. A. C. G., & Chen, K. (2025). Between innovation and standardization: Best practices and inclusive guidelines in computational communication science. *Journalism & Mass Communication Quarterly*, 102(1), 13–36. <https://doi.org/10.1177/10776990241301676>

Wissenschaftsrat. (2011). *Recommendations on Research Infrastructures in Humanities and Social Sciences*. https://www.wissenschaftsrat.de/download/archiv/10465-11_engl

Yehudi, Y., Cashman, M., Goedicke, M., Hasselbring, W., Katz, D. S., Müller, S., Goble, C., & Jay, C. (2025). Research software lifecycles and stages. *Electronic Communications of the EASST*, 85. <https://www.eceasst.org/index.php/eceasst/article/view/2693>

About the Authors



Christina Viehmann (Dr phil) is a senior researcher and head of the team Transparent Social Analytics at GESIS – Leibniz Institute for the Social Sciences. She is one of the service managers of the Methods Hub. Her research focuses on political communication, digital media environments, and NLP approaches to measure social phenomena.



Johannes Kiesel (PhD) is a senior researcher and lead of team Big Data Analytics at GESIS – Leibniz Institute for the Social Sciences, Cologne, Germany. He is one of the service managers of the Methods Hub. His research interests include conversational AI, human values, and computational reproducibility.



Arnim Bleier is a postdoctoral researcher at GESIS – Leibniz Institute for the Social Sciences. His research interests are in the field of computational social science, with an emphasis on reproducible research. In collaboration with social scientists, he develops models for the content, structure, and dynamics of social phenomena.

Chung-hong Chan (PhD, University of Hong Kong) is a researcher at GESIS. He writes R programs.



Po-Chun Chang is a research software engineer at GESIS – Leibniz Institute for the Social Sciences, Cologne, Germany. His interests include computational reproducibility and systems for web data analysis.



Raniere Gaia Costa da Silva is a research software engineer fighting the science reproducibility crisis.



Ahrabhi Kathirgamalingam is a postdoctoral researcher at GESIS – Leibniz Institute for the Social Sciences, Germany. Her research interests include phenomena such as racism, migration, and polarization in mediated discourses, as well as methodological questions relating to social bias in computational social science.



Taimoor Khan is a senior researcher at GESIS – Leibniz Institute for the Social Sciences, Cologne, Germany. His interests include reproducible computational workflows and web data analysis, especially regarding social dynamics.



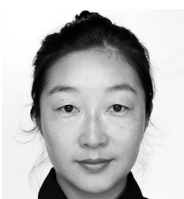
Stephan Linzbach is a researcher and data scientist at GESIS – Leibniz Institute for the Social Sciences. His work focuses on computational sociolinguistics and the capabilities of large language models to recall factual knowledge for knowledge graphs.



Fakhri Momeni is a researcher and data scientist at GESIS – Leibniz Institute for the Social Sciences. Her work focuses on computational social science, digital trace data, and the use of machine learning and large language models to study social and scientific communication and broader societal processes.



Felix Münch is a computational social scientist and serves as senior researcher and service manager for the Methods Hub at GESIS – Leibniz Institute for the Social Sciences. With a background in physics, journalism, and UX design, he focuses on network science, natural language processing, and the networked public sphere.



Ran Yu holds a PhD in computer science and is a senior researcher at GESIS – Leibniz Institute for the Social Sciences. Her research focuses on information retrieval, user modeling, knowledge graphs, and learning analytics. She is a member of the Methods Hub team and leads the DFG-funded EmergentIR project.



Stefan Dietze is professor of data and knowledge engineering at Heinrich-Heine-University Düsseldorf and scientific director of Knowledge Technologies for the Social Sciences (KTS) at GESIS. His research focuses on AI, natural language processing, information retrieval, and machine learning for mining and understanding large-scale Web and social media data.



Claudia Wagner is professor of applied computational social science at RWTH Aachen University and scientific director at GESIS. Her research focuses on understanding digital societies via digital behavioral data and AI methods. She develops research infrastructures and computational methods and is actively engaged in the international computational social science community.