Towards Transparency and Knowledge Exchange in AI-assisted Data Analysis Code Generation

Robert Haase[a,b,*]

[a]Data Science Center, Leipzig University, Humboldtstraße 25, 04105 Leipzig, Germany
[b]Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI) Dresden / Leipzig
ORCID (Robert Haase): https://orcid.org/0000-0001-5949-2327

Abstract.
Abstract will be added later

1 Introduction

Generative artificial intelligence (AI) and Large Language Models (LLMs) in particular are changing the way we do data science. Most prominently, scientists use the technology for interacting with scientific data [7], anwer data analysis questions [4, 5], generate data analysis code [8, 2, 1], and [re-]writing scientific manuscripts [6]. Unfortunately, the prompts sent to LLMs are commonly not stored, hindering knowledge exchange between scientists on how to use the technology efficiently and responsibly. At the time of publication of scientific code and manuscripts, the workflow that lead to a given set of data analysis code may be intransparent: It might be hard to identify the parts of the project which were implemented by a human and the parts that were created by AI. A professional peer-review system, for documenting how LLM-written code was prompted for, and which human reviewed it, is not established in contemporary scientific culture. However, such systems do exist for collaborative code editing involving multiple humans. E.g. the github.com online platform is well-established in the open-source software community for discussing issues and potential solutions, building code together, and for peer-reviewing it, before it is published to a wider community. As it was shown before that LLMs can solve real-world GitHub issues [3], developing an AI-assistant that inteacts with humans directly on the Github platform is the obvious next step. In this manuscript, I am presenting git-bob, a practical implementation of an LLM-based AI-assistant that can answer to GitHub issues, discuss potential solutions with humans iteratively, write code for them, and submit it as pull-request to reviewed by humans. It is technically similar to various online services for data analysis such as the OpenAI ChatGPT Data Analyst, with three major differences: 1) Multiple humans can interact with git-bob in one communication thread. This allows bringing together domain specialists, e.g. a life scientist, data-analyst and AI in one discussion, stimulating knowledge exchange. 2) Discussions with git-bob and resulting code-modifications are maintained in an online-platform that others can read and follow, making the interaction with the AI fully transparent, and 3) git-bob is open-source. As it uses a platform software developers and data analysts are used to, it does hardly change pre-existing workflows. It just allows to inject AI into discussions they have anyway. In a world where it was demonstrated that LLM-based systems can write entire papers, review and improve them [6] autonomously, such a solution is urgently need to be established as part of good scientific practice. The open source code of git-bob is available online and comes with detailed installation instructions so that everyone can start interacting with it on their github repositories: https://github.com/haesleinhuepf/git-bob

* Corresponding Author. Email: robert.haase@uni-leipzig.de

2 Features and limitations

A common workflow, demonstrated in Figure 1A, is that a user opens an issue on a repository on Github.com, where git-bob is installed. If the user is repository member, they can trigger git-bob to answer. If they are externals, a repository member has to do this. git-bob may then answer the question, potentially including a code snippet and resulting images. Users and the AI-assistant can then argue back and forth until some potential solution is reached. Optionally, git-bob can then be asked to submit a GitHub pull-request, e.g. with a Jupyter Notebook containing the entire solution to a given issue. A human would need to review this pull-request and merge it into the code base of the repository. Git-bob also has the capability to review pull-requests, e.g. originating from humans, but it is not allowed to merge them. This reflects established practices in science, where eventually a human is reponsible for data anaylsis code that becomes part of the project. Additional tasks git-bob is capable of are: 1) The assistant can support users of open source libraries by providing advice and code examples,as shown in Supplementary Figure S1. In case the assistant is not sure about the answer, it is capable of forwarding the question to a human, as shown in Supplementary Figure S2. 2) It can write and execute data analysis code, e.g. to summarize and plot CSV files which are stored in a repository, as demonstrated in Supplementary Figure S4. When analysing data, the assistant is intrinsically limited by the capabilities of the used LLM. For example, it has been shown before that common commercial state-of-the-art LLMs can solve bio-image analysis questions by generating functionally correct code just above 50% of tested cases [2]. This fundamental limitation may disappear when improved LLMs are published. For now, it can be evaded in multi-turn interactions between humans and AI. Yet, we humans need to guide the AI towards a workable solution.

A highlight of git-bob is that a local installation or an institutional internet server are not required. Git-bob is implemented as GitHub Workflow, and hence runs within the IT infrastructure of github.com.

Figure 1. example interaction with git-bob - figure placeholder

It is compatible with GPT4-omni, other OpenAI LLMs, Anthropic's Claude, Google Gemini, models hosted on Github Models Marketplace and other models which use the OpenAI application programming interface (API). Git-bob reports which model was used in all of its messages, as good scientific practice suggests. The first three of the mentioned vendors require payment for using their services through the API, the others may offer usage of their API for free. Obviously, the communication with the selected LLM is transmitted to the service provider, including source code files from the repository and images provided with the github issue. Hence, users are recommended to not submit any personal or sensitive information.

Using git-bob and also LLMs in general does not eliminate the need for expertise in the domain we are working in. For example, when tasked to setup a bio-image analysis workflow, at least one of the the humans interacting with git-bob should have the expertise to judge if a proposed solution is reasonable at least. When LLMs improve, especially their reasoning capabilities, this limitation may feel less and less relevant. Eventually, as correct results are commonly unknown in data analysis research projects, a human may still be required who has the right expertise to judge if a workflow is producing correct results.

Git-bob can be used in private repositories giving scientists the necessary privacy to work on projects before they eventually publish their work. For example, this manuscript was edited with AI-assistance in a private repository as shown in Supplementary Figure **??**.

3 Conclusion

LLMs are being integrated in contemporary scientific workflows unavoidably, but documentation of how in detail they are imployed is commonly not done, also because of lack of tools allowing this conveniently. If the scientific community documented usage of LLM prompts like they document usage of open source data analysis libraries, we could learn from each other how to prompt efficiently and responsibly. To overcome current limitations, I propose git-bob for documenting interactions with LLM-based AI. It works integrated on github.com enabling scientists to interact with an LLM via Github Issues and Pull-Requests offering new ways for implementing good scientific practice for the documentation of discussions between humans and AI when they are working on projects together.

Acknowledgements

RH acknowledges the financial support by the Federal Ministry of Education and Research of Germany and by Sächsische Staatsministerium für Wissenschaft, Kultur und Tourismus in the programme Center of Excellence for AI-research "Center for Scalable Data Analytics and Artificial Intelligence Dresden/Leipzig", project identification number: ScaDS.AI.

References

[1] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. de Oliveira Pinto, et al. Evaluating large language models trained on code. *CoRR*, abs/2107.03374, 2021. URL https://arxiv.org/abs/2107.03374.

[2] R. Haase, C. Tischer, J.-K. Hériché, and N. Scherf. Benchmarking large language models for bio-image analysis code generation. *bioRxiv*, 2024. doi: 10.1101/2024.04.19.590278. URL https://www.biorxiv.org/content/early/2024/04/25/2024.04.19.590278.

[3] C. E. Jimenez, J. Yang, A. Wettig, S. Yao, K. Pei, O. Press, and K. Narasimhan. Swe-bench: Can language models resolve real-world github issues?, 2024. URL https://arxiv.org/abs/2310.06770.

[4] Y. Lai, C. Li, Y. Wang, T. Zhang, R. Zhong, L. Zettlemoyer, S. W. tau Yih, D. Fried, S. Wang, and T. Yu. Ds-1000: A natural and reliable benchmark for data science code generation, 2022.

[5] W. Lei, C. Fuster-Barceló, G. Reder, et al. Bioimage.io chatbot: a community-driven ai assistant for integrative computational bioimaging. *Nature Methods*, 21:1368–1370, 2024. doi: 10.1038/s41592-024-02370-y.

[6] C. Lu, C. Lu, R. T. Lange, J. Foerster, J. Clune, and D. Ha. The AI Scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.

[7] L. A. Royer. The future of bioimage analysis: a dialog between mind and machine. *Nature Methods*, 20(7):951–952, 2023. ISSN 1548-7105. doi: 10.1038/s41592-023-01930-y. URL https://doi.org/10.1038/s41592-023-01930-y.

[8] L. A. Royer. Omega – harnessing the power of large language models for bioimage analysis. https://doi.org/10.5281/zenodo.8240289, 2023.

4 Supplementary material

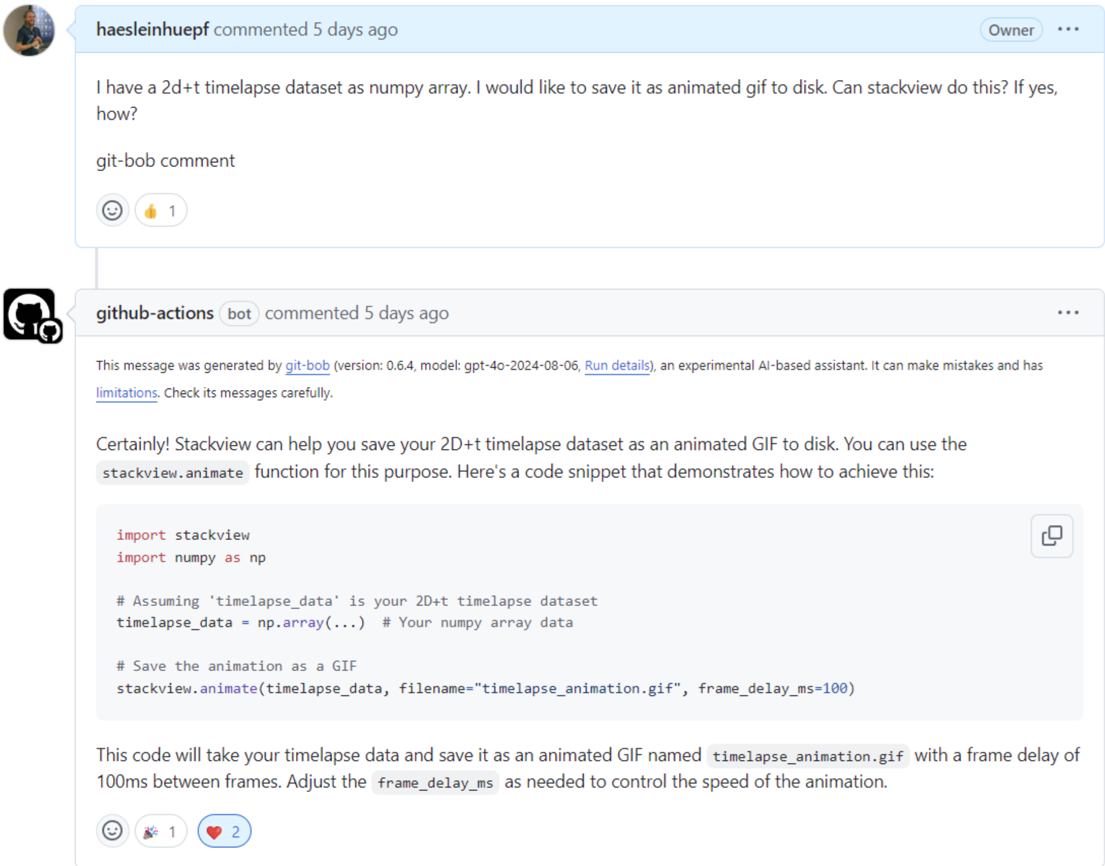

Figure S1. Use-case example for supporting users: The assistant can be configured to act as expert on a specific Python library and answer user questions. The entire discussion and corresponding code can be read online: https://github.com/haesleinhuepf/stackview/issues/79

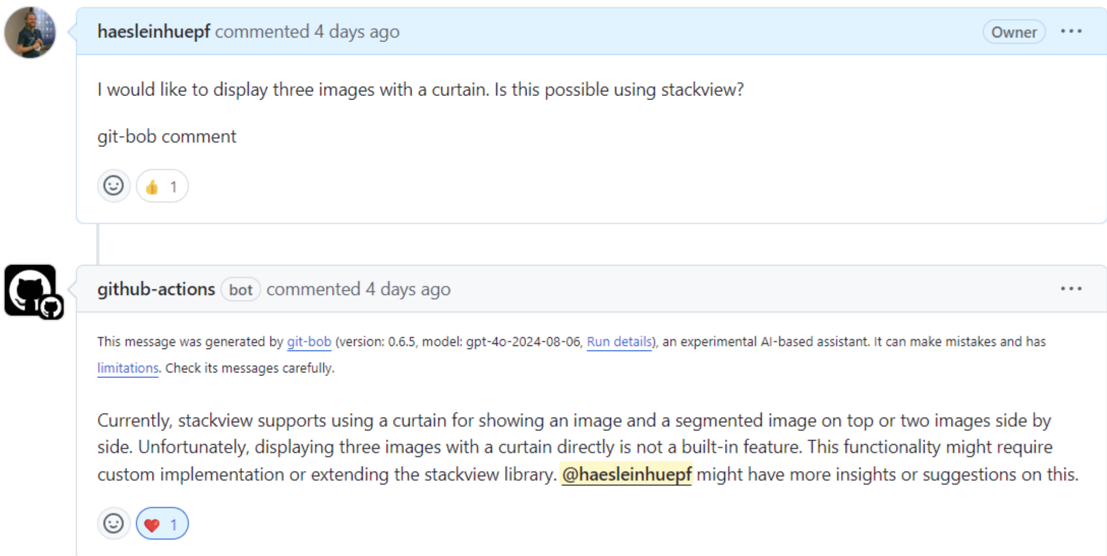

Figure S2. Use-case example for asking an expert: The answer to the question shown here is "No", but this is nowhere written in the documentation or the configuration of the assistant. In this case the assistant is not sure, and it can be configured to forward a question to a maintainer of the library where the question arrived. The entire discussion and corresponding code can be read online: https://github.com/haesleinhuepf/stackview/issues/80

Figure S3. Use-case example for plotting data: After explaining the assistant the folder structure of the project, it generates code for parsing a folder of CSV files and plotting results. The entire discussion and corresponding code can be read online: https://github.com/NFDI4BIOIMAGE/training/issues/250

Figure S4. Use-case example for working on scientific manuscripts: After a first draft of the manuscript was written, git-bob was asked to formulate an abstract. The abstract was added to the .tex document after git-bob sent a pull-request and the human author reviewed it. The entire discussion can be read online: https://github.com/haesleinhuepf/git-bob-manuscript/issues/